

The Shape of Football: Applying Dimensionality Reduction Algorithms and Topological Data Analysis to Player Analytics

[**Martin Alejandro Arce Llobera**]

A thesis submitted for the degree of

Master of Mathematical Sciences (Advanced)

The Australian National University

Mathematical Sciences Institute



May 2024

© Copyright by [Martin Alejandro Arce Llobera], 2024

All Rights Reserved

To my mouse, family and friends

Disclaimer

I hereby declare that the work undertaken in this thesis has been conducted by me alone, except where indicated in the text. I conducted this work between August 2023 and May 2024, during which period I was a Master's student at the Australian National University. This thesis, in whole or any part of it, has not been submitted to this or any other university for a degree.

This thesis has been compiled as a Thesis by Compilation in accordance with relevant ANU policies.

A handwritten signature in black ink, appearing to read "MA" followed by a stylized surname.

[Martin Alejandro Arce Llobera]

23 May 2024

Acknowledgements

First, I would like to thank the authors of the blogs Math3ma, John Baez and ncatlab, as well as the creators of the Youtube channels Harpreet Bedi, Daniel Tubbenhauer, Borcherds, Fematika, for their fabulous content, who helped me understand minor yet important details. Also, to Adele Jackson for her notes on UMAP which shed a light in an initially dark forest.

Second, I would like to thank Professor Anand Deopurkar for his help throughout the Master's degree.

Third, I am indebted to my friends Juan and Seba, for always motivating me to learn Mathematics and for sharing their wisdom.

Fourth, I would like to thank my parents and my love for their endless support.

Finally, I would like to thank my supervisors James and Kate who have made this thesis an incredibly fun journey.

Abstract

In this thesis, we will explore football (soccer) analytics from a mathematical point of view. We will approach this matter using dimensionality reduction algorithms, such as UMAP, with a comprehensive explanation of its underlying mathematical ideas being a key contribution of this work. UMAP will help reduce the dimensionality of our high-dimensional football dataset, which will then be clustered using the Mapper algorithm to enhance visualization. Moreover, we will also apply persistent homology, the main topological data analysis tool, to extract astonishing football insights.

Contents

List of Figures	iv
0 Introduction	1
0.1 Thesis outline	4
1 Background knowledge	6
1.1 General background	6
1.2 Geometry background	7
1.3 Topological data analysis background	10
1.4 Category theory background	27
1.5 Background on selected optimization methods	49
2 Dimensionality reduction algorithms	54
2.1 PCA	55
2.2 UMAP	61
2.3 PCA and UMAP applications	75
2.4 Mapper	79
3 Football applications	82
4 Conclusions	93
Bibliography	95

List of Figures

1	Data points in \mathbb{R}^2 that resemble a koala.	2
2	Four identical sets when examined using simple summary statistics, but considerably different when graphed, Schutz (2010).	3
1.1	S^2	7
1.2	C	8
1.3	Three simplicial complexes: three points, a hollow triangle, and a hollow tetrahedron.	23
2.1	Two cameras recording the position of the yellow goalkeeper in a foosball game, image by Freepik (with modifications).	55
2.2	The dataset has been sampled with noise from a sine wave, McInnes (2018).	62
2.3	Open cover of the dataset, McInnes (2018).	62
2.4	Čech complex of the dataset, McInnes (2018).	63
2.5	Open cover of the dataset considering a specific geodesic distance, McInnes (2018).	64
2.6	Overlapping graphs of finite fuzzy simplicial sets, McInnes (2018).	71
2.7	Fuzzy union of finite fuzzy simplicial sets, McInnes (2018).	72
2.8	100 images of the MNIST dataset, LeCun et al. (2010).	75
2.9	2-dimensional PCA applied to the MNIST dataset.	76
2.10	2-dimensional UMAP applied to the MNIST dataset.	76
2.11	100 images of the Fashion-MNIST dataset, Xiao et al. (2017).	77
2.12	2-dimensional PCA applied to the Fashion-MNIST dataset.	78
2.13	2-dimensional UMAP applied to the Fashion-MNIST dataset.	78
2.14	Dataset of a square with noise.	79
2.15	Projecting the dataset into the y coordinate and defining an open cover represented with colors.	80

2.16	Clusters of the preimages after applying some clustering algorithm.	80
2.17	Nerve of the open cover.	80
3.1	Correlation matrix of our football dataset.	83
3.2	Correlation matrix of our football dataset after applying the Matrix Sorting Algorithm.	85
3.3	Sorted correlation matrix of our football dataset after deleting highly correlated variables.	86
3.4	2-dimensional PCA applied to our football dataset.	87
3.5	2-dimensional UMAP applied to our football dataset.	87
3.6	Football dataset after applying Mapper in combination with UMAP and agglomerative clustering.	89
3.7	Box plots of the <i>ST</i> of the best 15 teams and of the bottom 15 teams. . .	91
3.8	FC Barcelona histogram of normalized ST when adding a new player from LaLiga.	92

Introduction

*"If people do not believe that mathematics is simple,
it is only because they do not realize
how complicated life is."*

John von Neumann

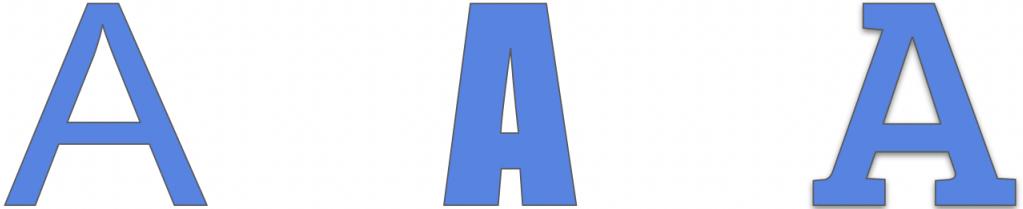
Mathematics and football (soccer) lie at opposite extremes of the abstraction spectrum. The precise and methodical discipline on one side, and the visceral and emotionally charged game on the other. Being raised in a country like Argentina, where God is a synonym for Diego Maradona, one of our most celebrated players, and having always enjoyed mathematics, have made me appreciate these polar frameworks. It is in this context that I naturally came up with the idea of building a bridge between mathematics and football.

Books like Moneyball (Lewis (2004)), also produced as a Hollywood movie, have publicly shown the power of applying statistics to baseball, and there are hundreds of articles discussing how basketball has changed since the introduction of analytics tools. Nonetheless, although football is the world's most played sport and there is a massive industry behind it, analyzing this particular game through a mathematical lens is still in its early days.

In the current era, where the mighty machine learning (ML) captures most of the attention of data science projects, topological data analysis (TDA) has started to penetrate and gain more and more adepts. Moreover, the combination of ML and TDA has begun to gain traction, as seen in articles such as Hofer et al. (2017), and Love et al. (2021). If data is scarce and it is not possible to generate synthetic one or it is too expensive, then adding robust topological features can increase a ML model's predictive power, as seen in Zhang et al. (2021). But what exactly is TDA? To answer this question, we first need to understand topology and why it is useful.

We can describe topology as the mathematical discipline that lives in a world made of clay. In this world, distances are not important and objects can be continuously deformed yet they preserve their fundamental properties. Although this setting appears to be rather

abstract, our brains are brilliant at using some sort of topology. For example, we easily identify the following three drawings as the same letter:



Hence, topology is a remarkable tool in helping us understand what the foundational characteristics of an object are. Leveraging these ideas we find TDA, a nascent field that merges data science with topology, whose mantra is “Data has shape”. In order to easily understand this motto, we have created “The Kodata”:

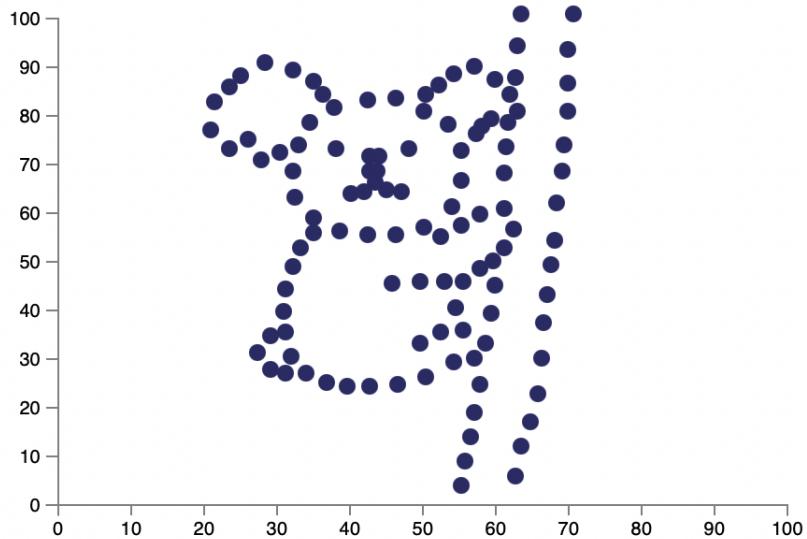


Figure 1: Data points in \mathbb{R}^2 that resemble a koala.

There are different ways of describing The Kodata. Some may say it is just a subset of \mathbb{R}^2 with cardinality 115. Others might refer to it using common summary statistics, such as having a mean equal to $(48.00, 57.95)$ and a standard deviation of $(13.59, 23.83)$, where each axis is considered separately. However, the limitations of these points of view were first observed in Anscombe (1973). In this paper, Anscombe showed that four highly distinct datasets can share several summary statistics. We visualize this situation with the following picture:

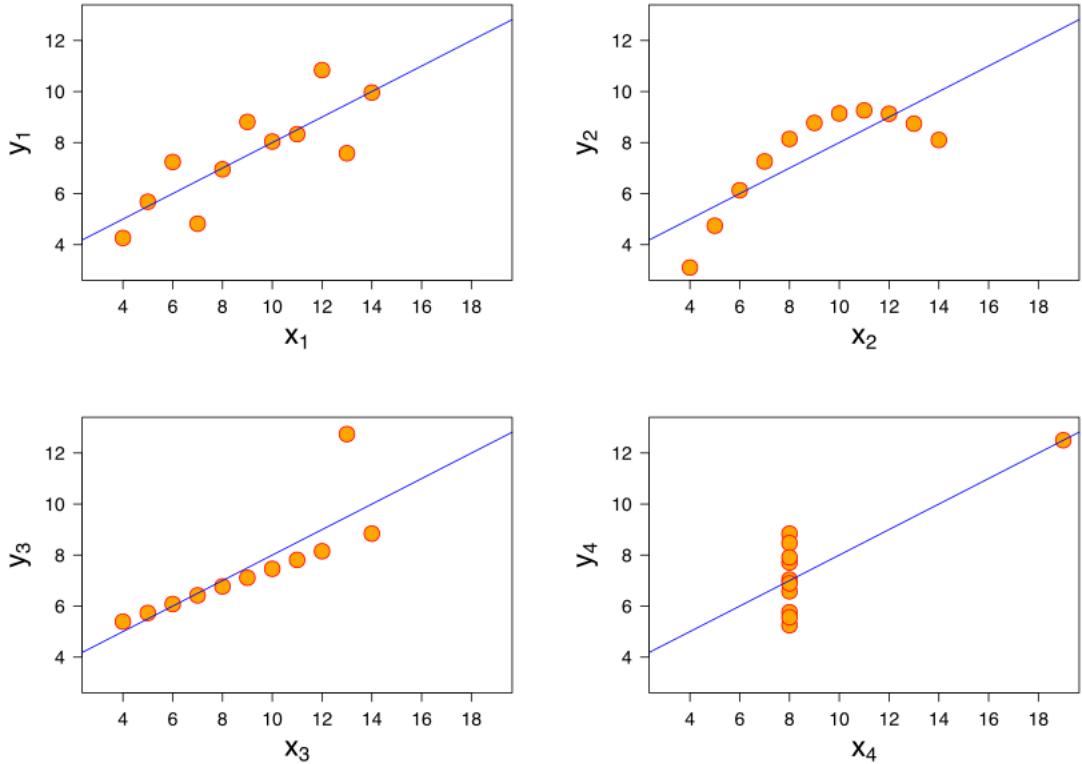


Figure 2: Four identical sets when examined using simple summary statistics, but considerably different when graphed, Schutz (2010).

This simple illustration enlightens the importance of data visualization. Nevertheless, there are plenty of other reasons to embrace accurate visualization techniques. For instance, domain experts can obtain plenty of insights from appropriate graphs.

Another rationale for using visualization is explainability. Several ML models are as powerful as they are mysterious. Combining these methods with TDA and visualization can add an important layer of understanding. Furthermore, we dream of a future where artificial intelligence models synergize with humans. Thus, understanding these tools is crucial for moving forward on this path.

Before moving on to the specific content of this thesis, we would like to remark that no human-made model is perfect. Given that reality is arguably the only perfect design, the correct question we should be asking ourselves is if our model captures the notions we are interested in.

As is often the case in real-world applications, we first conducted countless experiments with our football dataset. After identifying the appropriate tools, we delved into fully understanding them. With this in mind, we proceed to communicate what lies in this dissertation.

0.1 Thesis outline

This thesis focuses on dimensionality reduction algorithms and topological tools, since these are the instruments that will allow us to obtain numerous insights from our football dataset. Thus, the presentation of the chapters will be oriented towards these tools.

The first chapter consists of the background knowledge, where we will find five sections. In the initial section, we will discuss a few basic and disconnected concepts. This is meant to be a section that contains every notion needed that does not fit in other modules. In the second section, we will talk about a small number of geometric ideas that will be crucial to understanding UMAP, our main dimensionality reduction algorithm, which is an example of a “manifold learning algorithm”. In the third section, we will formalize and expand on the notions of topological data analysis we have already discussed. In the fourth one, we will spend a large amount of time discussing category theory. This mathematical area is key for understanding UMAP, since this algorithm is based on categorical ideas. In the fifth and last section, we will introduce some optimization methods, needed for cleaning our dataset. For completeness, we will provide proofs for some of the most important theorems throughout this chapter.

The second chapter corresponds to three dimensionality reduction algorithms: PCA, UMAP, and Mapper. Over the course of this chapter, our focus will be on the mathematical foundations of these algorithms, rather than on their implementation details. PCA, given its simplicity, acts as a first step in almost any data science project. Exploring UMAP will be one of the most important contributions of this thesis. UMAP is considered one of the state-of-the-art dimensionality reduction algorithms. In order to understand how the author developed this tool, knowledge in many mathematical areas is required. This makes UMAP difficult to fully comprehend, even for specialized audiences. Although the ideas behind UMAP are outstanding and it produces excellent results, we will see that there is a clear disconnection between the mathematics that inspired UMAP and the actual algorithm. We will try to explain each theoretical detail of UMAP in a self-contained manner and make adjustments where necessary. While there have been several attempts to explain different aspects of UMAP, we believe our work provides a comprehensive exposition of its mathematical ideas. The last algorithm we will discuss is a simple one called Mapper. It constitutes a fantastic tool to enhance visualization and to leverage the strengths of PCA and UMAP. It should be noted that there are many other dimensionality reduction algorithms, such as t-SNE and LargeVis. During the experimentation process, we tested many of these algorithms on our dataset. Since UMAP produced the best results, we decided to dig into its details.

The third chapter will concern the application of our techniques to football analytics. We

will employ all the discussed tools to obtain remarkable insights. It is important to keep in mind that our main goal is to understand our dataset, highlighting points of interest which will let us identify what the important questions to ask are.

Finally, in the conclusion section we will summarize our contributions and discuss future lines of research.

It is time now to move on to the first chapter.

Background knowledge

1.1 General background

“My brain is open.”

Paul Erdős

When explaining UMAP, we will see that the main objects we are going to be dealing with are called finite fuzzy simplicial sets and finite extended-pseudo-metric spaces. We will use this brief section just to introduce the concepts of fuzzy sets, and extended-pseudo-metric spaces along with their ordinary maps. We start with our first definition:

Definition 1.1.1. A **fuzzy set** (S, μ) consists of a set S and a membership function $\mu : S \rightarrow [0, 1]$. The value $\mu(x)$ for $x \in S$ represents the degree of membership of x in (S, μ) .

Fuzzy sets generalize the classic binary notion of membership of sets. Notice that classical sets can be seen as fuzzy sets where μ can only take the values 0 or 1.

Definition 1.1.2. An **extended-pseudo-metric space** (X, d) is a set X and a map $d : X \times X \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$ such that:

1. $x = y$ implies $d(x, y) = 0$;
2. $d(x, y) = d(y, x)$;
3. $d(x, z) \leq d(x, y) + d(y, z)$.

The word “extended” implies that we are allowing ∞ as a possible value. The word “pseudo” implies that distinct $a, b \in X$ may satisfy $d(a, b) = 0$.

Definition 1.1.3. Let $(X, d_X), (Y, d_Y)$ be extended-pseudo-metric spaces. We call a map $f : X \rightarrow Y$ **non-expansive** if for any $a, b \in X$ we have

$$d_X(a, b) \geq d_Y(f(a), f(b)).$$

Having established our basic definitions, we proceed to the next background section.

1.2 Geometry background

“Manifolds crop up everywhere in mathematics.”

John M. Lee

UMAP is a member of a class of methods called “manifold learning algorithms”. The main characteristic of this class is that it assumes high-dimensional datasets can be approximated as a lower-dimensional manifold embedded in a higher-dimensional space. This means datasets can be thought of as samples from a distribution whose support is a lower-dimensional manifold. In this short section, we will define some geometric notions that will be useful when dealing with UMAP. To achieve this, we will make use of Andrews (2017). We proceed with our explanation:

We are particular interested in Riemannian manifolds. Before introducing them, let us recall some basics of geometry along with a couple of examples.

A **manifold** \mathcal{M} is a topological space that locally resembles Euclidean space. A **smooth manifold** can be described as a manifold \mathcal{M} that is suited for applying differential calculus techniques. In order to improve our intuition of these objects, consider the following cases which are easy to visualize:

Example 1.2.1. The 2-sphere $S^2 = \{(x, y, z) \in \mathbb{R}^3 : \|(x, y, z)\| = 1\}$ is a smooth manifold.

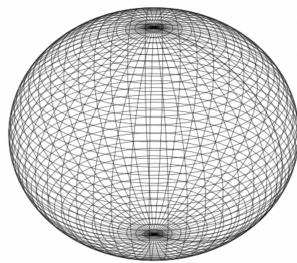
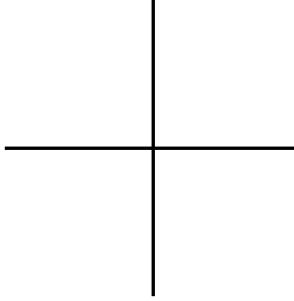


Figure 1.1: S^2 .

Non-example 1.2.2. We proceed to prove that the cross, shown in Figure 1.2,

$$C = \{(x, 0) \in \mathbb{R}^2 : -1 \leq x \leq 1\} \cup \{(0, y) \in \mathbb{R}^2 : -1 \leq y \leq 1\}$$

is not a manifold.

Figure 1.2: C .

Proof. Suppose C locally resembles the Euclidean space of dimension $n > 0$ at the intersection point $(0, 0)$. This means that, for some $\epsilon > 0$, there exists a homeomorphism $f : U \rightarrow B(0, \epsilon)$ such that $(0, 0) \mapsto 0 \in \mathbb{R}^n$, where $U \subseteq C$ is a neighbourhood of $(0, 0)$ and $B(0, \epsilon) = \{x \in \mathbb{R}^n : |x| < \epsilon\}$. We will now consider the two possible cases:

$n = 1$: consider the restriction $f|_{U - \{(0, 0)\}} : U - \{(0, 0)\} \rightarrow B(0, \epsilon) - \{0\} \subseteq \mathbb{R}$. Notice that $U - \{(0, 0)\}$ has four connected components whereas $B(0, \epsilon) - \{0\}$ only two.

$n > 1$: consider the restriction $f|_{U - \{(0, 0)\}} : U - \{(0, 0)\} \rightarrow B(0, \epsilon) - \{0\} \subseteq \mathbb{R}^n$. Notice that $U - \{(0, 0)\}$ has four connected components whereas $B(0, \epsilon) - \{0\}$ is connected.

Since homeomorphisms preserve the number of connected components, we arrive at a contradiction. Therefore, the cross C is not a manifold.

□

We can now present the object we will be using when discussing our most important dimensionality reduction algorithm:

Definition 1.2.3. Let \mathcal{M} be a smooth manifold. A **Riemannian metric** g on \mathcal{M} is a smoothly chosen inner product $g_x : T_x \mathcal{M} \times T_x \mathcal{M} \rightarrow \mathbb{R}$ on each of the tangent spaces $T_x \mathcal{M}$ of \mathcal{M} . In other words, for each $x \in \mathcal{M}$, $g = g_x$ satisfies:

1. $g(u, v) = g(v, u)$ for all $u, v \in T_x M$;
2. $g(u, u) \geq 0$ for all $u \in T_x M$;
3. $g(u, u) = 0$ if and only if $u = 0$.

Furthermore, g is smooth in the sense that for any smooth vector fields X and Y , the function $x \mapsto g_x(X_x, Y_x)$ is smooth.

Example 1.2.4. The standard inner product on the Euclidean space \mathbb{R}^n is a classic example of a Riemannian metric.

An important result in differential geometry is that every smooth manifold admits a Riemannian metric, and in fact, more than one. Thus, we can always add this extra structure.

Riemannian manifolds enrich the structure of smooth manifolds, allowing us to compute angles, volumes, and distances between points. When exploring UMAP, we will be particularly interested in approximating the geodesic distance between different points. The **geodesic distance** is defined as the length of the shortest curve between two points within the manifold \mathcal{M} .

We now move on to the next background section.

1.3 Topological data analysis background

“Data has shape and the shape matters.”

Gunnar Carlsson

Topological data analysis is a mathematical field that combines topology with data science, offering powerful perspectives in understanding data.

We will split this section into two parts. The first will introduce the basic concepts of this mathematical discipline and will conclude with a highly important theorem, whereas the second will discuss the main tool of topological data analysis: persistent homology.

This section is heavily influenced by Nanda (2021).

We now proceed to the first subsection.

1.3.1 Towards The Nerve Theorem

In this subsection we will define all the necessary concepts that will help us grasp The Nerve Theorem. We will see that this theorem is a stepping stone to being able to extract useful and mathematically justified knowledge from datasets. We start with our first definition:

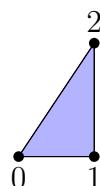
Definition 1.3.1. Let V be a finite non-empty set whose elements are called vertices. A **simplicial complex on V** is a collection K of non-empty subsets of V which satisfies:

- For each $v \in V$, the singleton set $\{v\} \in K$.
- If $\tau \in K$ and $\sigma \subseteq \tau$, then $\sigma \in K$.

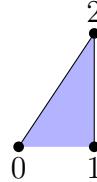
The elements of K are called **simplices**.

Example 1.3.2. Let $V = \{0, 1, \dots, n\}$ be a set of vertices. The **solid n -simplex $\Delta(n)$** is the simplicial complex defined on V such that every non-empty subset of V constitutes a simplex.

For $n = 2$, we have $\Delta(2) = \{\{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}\}$. Visually, this is



Non-example 1.3.3. The collection $M = \{\{0\}, \{1\}, \{2\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}\}$ is not a simplicial complex. This is because it is missing the subset $\{0, 1\}$ but including $\{0, 1, 2\}$. Visually, this is



Simplicial complexes are a fantastic tool for representing topological spaces. Instead of having to deal with the geometric aspects of them, which can be hard to handle, we can convert topological spaces into simple combinatorial objects. One of the main objectives of this section will be to explain how to perform this conversion which, in order to be faithful, will have to preserve the topological structure of the space.

Let us now delve into definitions related to simplicial complexes:

Definition 1.3.4. Let K be a simplicial complex and let $\tau, \sigma \in K$ be two simplices. We say τ is a **face of σ** if each vertex $v \in \tau$ is also a vertex of σ . We denote this as $\tau \leq \sigma$.

Definition 1.3.5. Let K be a simplicial complex. A **subcomplex L of K** is a subset $L \subseteq K$ which satisfies:

- Let $\sigma \in L$. If τ is a face of σ in K , then $\tau \in L$.

Definition 1.3.6. Let K, L be simplicial complexes and let K_0, L_0 be their respective sets of vertices. A **simplicial map** $f : K \rightarrow L$ is an assignment $K_0 \rightarrow L_0$ of vertices to vertices which sends simplices to simplices. This means, for each simplex $\sigma = \{v_0, v_1, \dots, v_k\} \in K$, its image $f(\sigma) = \{f(v_0), f(v_1), \dots, f(v_k)\}$ is a simplex of L .

It is often useful to connect abstract simplicial structures with Euclidean spaces. In order to do that, we require the following definition:

Definition 1.3.7. Let K be a simplicial complex and let K_0 be the vertices of K . Let $\phi : K_0 \rightarrow \mathbb{R}^n$ be any function. The **geometric realization of K** with respect to ϕ is

$$|K|_\phi := \bigcup_{\sigma \in K} |\sigma|_\phi$$

where, for each $\sigma = \{v_0, \dots, v_k\}$ in K , the set $|\sigma|_\phi \subseteq \mathbb{R}^n$ is the subset of \mathbb{R}^n given by

$$\left\{ \sum_{i=0}^k t_i \phi(v_i) : \text{ where } t_i > 0 \text{ and } \sum_{i=0}^k t_i = 1 \right\} .$$

We call ϕ an **affine embedding** of K in \mathbb{R}^n if it is injective.

The following proposition will make this transition less constrained:

Proposition 1.3.8. For any two affine embeddings $\phi, \psi : K_0 \rightarrow \mathbb{R}^n$ there is a homeomorphism $|K|_\phi \simeq |K|_\psi$ between the corresponding geometric realizations.

Proof. Refer to Nanda (2021), Proposition 1.8. □

Because of this result, we will write $|K|$ for the geometric realization of a simplicial complex K , omitting the affine embedding sub-index.

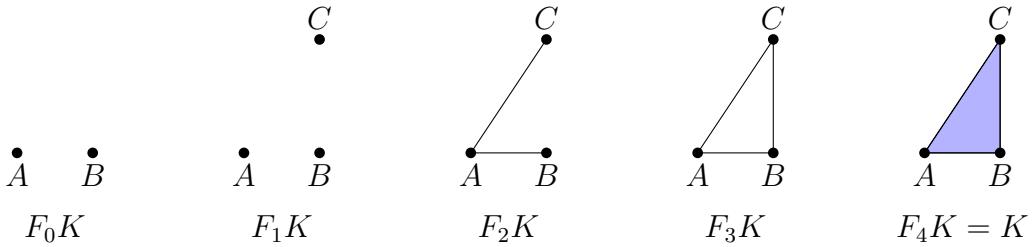
A critical aspect in TDA involves understanding how the data topological structure evolves over some parameter. This takes us to the concept of filtration:

Definition 1.3.9. Let K be a simplicial complex. A **filtration of K** is an increasing family of simplicial complexes $\{F_i K\}_{i=0}^n$, such that $F_n K = K$ for some integer n . In other words, it is a nested sequence

$$F_0 K \subseteq F_1 K \subseteq \cdots \subseteq F_{n-1} K \subseteq F_n K = K$$

where $F_i K$ is a subcomplex of $F_j K$ for $i \leq j$.

Example 1.3.10. Let $K = \{\{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}$ be a simplicial complex. The following is the illustration of a particular filtration of K :



Although we all have some kind of definition of the word “dataset” in our minds, a precise topological definition is necessary for our purposes:

Definition 1.3.11. A **dataset** is a metric space (S, d) , where S is a finite set and d is a metric on S . In general, it is going to consist of a finite subset of \mathbb{R}^n along with the restriction of the standard Euclidean metric.

It is important to remark that, in other sections of this thesis, the word “dataset” will also be used to refer just to the finite set of points S , instead of the metric space (S, d) . This distinction will be clear from the context and will not affect our understanding.

With this accurate topological definition, we can now explore, in two different ways, how to construct simplicial complexes from data.

Definition 1.3.12. Let (S, d) be a dataset. The **Vietoris-Rips filtration of S** is an increasing family of simplicial complexes $VR_\epsilon(S)$ indexed by real numbers $\epsilon \geq 0$, defined as follows:

- A subset $\{x_0, x_1, \dots, x_k\} \subseteq S$ forms a k -dimensional simplex in $VR_\epsilon(S)$ if and only if the pairwise distances satisfy $d(x_i, x_j) \leq \epsilon$ for all i, j .

This implies that a set of points produces a simplex if every pair of points is within a distance smaller than ϵ .

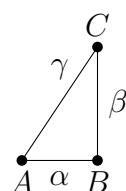
Definition 1.3.13. Let (S, d) be a dataset and let $S \subseteq \mathbb{R}^n$. The **Čech filtration of S** with respect to \mathbb{R}^n is an increasing family of simplicial complexes $C_\epsilon(S)$ indexed by $\epsilon \geq 0$, defined as follows:

- A subset $\{x_0, x_1, \dots, x_k\} \subseteq S$ forms a k -dimensional simplex in $C_\epsilon(S)$ if and only if there exists some $r \in \mathbb{R}^n$ satisfying $d(r, x_i) \leq \epsilon$ for all i .

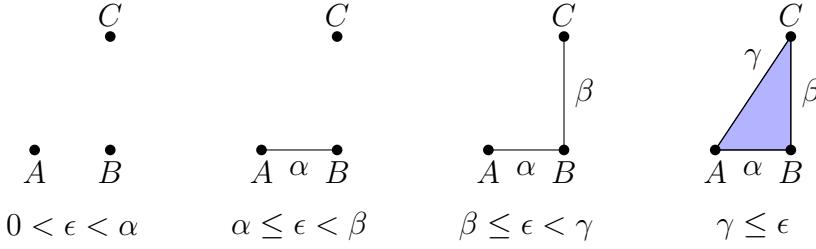
This means that, for a set of points to make a simplex, the intersection of balls of size ϵ around each point is non-empty.

The simplest way to illustrate the differences between these two filtrations is through a visual example:

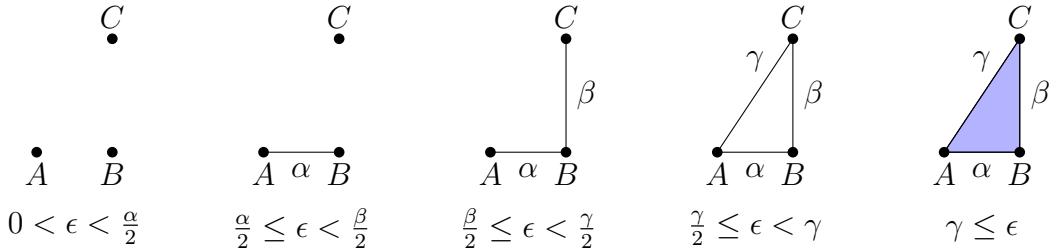
Example 1.3.14. Let $S = \{A, B, C\} \subseteq \mathbb{R}^2$ and $\alpha = d(A, B), \beta = d(B, C), \gamma = d(A, C)$, such that $0 < \alpha < \beta < \gamma$, where d is the Euclidean metric. Visually, this is



Vietoris-Rips filtration of S :



Čech filtration of S :



The Vietoris-Rips filtration is computationally more efficient to obtain because calculating pairwise distances is usually less computationally intensive than computing the intersection of multiple balls. However, the Čech filtration is seen as more accurate in capturing the topological structure of the dataset. This will become clear once we have seen Theorem 1.3.29 and its Corollary 1.3.30. Before introducing these, we need to first understand some essential algebraic topology concepts.

Definition 1.3.15. Let X and Y be topological spaces and let $f, g : X \rightarrow Y$ be continuous functions. We say f and g are **homotopic** if there exists a continuous function

$$F : X \times [0, 1] \rightarrow Y \text{ such that } F(x, 0) = f(x) \text{ and } F(x, 1) = g(x), \forall x \in X.$$

We denote this as $f \simeq g$. We call F a **homotopy** from f to g .

Example 1.3.16. Let $f, g : \mathbb{R} \rightarrow \mathbb{R}$ be any continuous functions. Let us define the function $F : \mathbb{R} \times [0, 1] \rightarrow \mathbb{R}$ as

$$F(x, t) = (1 - t)f(x) + tg(x).$$

Notice that F is continuous since it is a sum and a product of continuous functions. Moreover, we have $F(x, 0) = f(x)$ and $F(x, 1) = g(x)$. Therefore, $f \simeq g$.

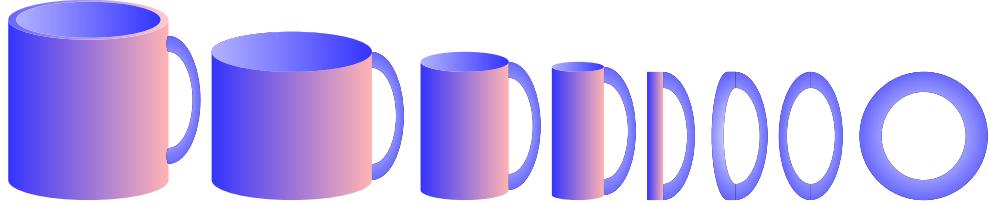
Definition 1.3.17. Let X and Y be topological spaces. We say that X and Y are **homotopy equivalent** if there exist continuous maps $f : X \rightarrow Y$ and $g : Y \rightarrow X$, such that:

- $f \circ g$ is homotopic to the identity map on Y ;
- $g \circ f$ is homotopic to the identity map on X .

We denote this as $X \simeq Y$.

Homotopy equivalent spaces exhibit similar topological structure, as seen, for example, by the fact that they have isomorphic homology groups. Intuitively, homology groups tell us about the number of “holes” and the number of connected components of a topological space. One kind of “hole”, in this context, can be described as an empty section enclosed by a loop. For instance, a circle has a “hole”. We often informally say that two topological spaces are homotopy equivalent if we can continuously shrink or stretch one into the other, while preserving the number of connected components and “holes”.

Example 1.3.18. An intuitive homotopy visualization: from a mug to a doughnut.



Bombadil (2013).

Definition 1.3.19. We say two simplicial complexes K and L are **homotopy equivalent** if their geometric realizations $|K|$ and $|L|$ are homotopy equivalent, in the sense of the above definition.

Definition 1.3.20. Let X be a topological space. We say X is **contractible** if the identity map $1_X : X \rightarrow X$ where $1_X(x) = x$ is homotopic to a constant map. We say X is **locally contractible** if every $x \in X$ has arbitrarily small open contractible neighbourhoods.

Intuitively, a topological space X is contractible if it can be shrunk to one of its points.

Example 1.3.21. Any non-empty convex subset X of \mathbb{R}^n , endowed with the subspace topology, has the property of being contractible. This can be seen by taking any point $p \in X$, the constant function $f_p : X \rightarrow X$ as $f_p(x) = p$, and $F : X \times [0, 1] \rightarrow X$ as

$$F(x, t) = (1 - t)1_X(x) + tf_p(x).$$

We see that $F(x, 0) = 1_X(x)$ and $F(x, 1) = f_p(x)$. Hence, X is contractible. It is important to remark that F is a valid homotopy because X is convex, which implies that the image of F is indeed in X .

In particular, the solid n -simplex $\Delta(n)$ is contractible because of the convexity of its geometric realization.

We are now getting close to The Nerve Theorem, our main result of this subsection. In order to discuss it, we first need to know what the nerve of an open cover is. Before providing this definition, we remind ourselves of a basic topological definition, essential for understanding the concept of nerve that follows.

Definition 1.3.22. Let X be a topological space and let A be some finite index set. A **finite open cover U_\bullet of X** is a collection of open subsets $U_\alpha \subseteq X$ satisfying

$$X = \bigcup_{\alpha \in A} U_\alpha.$$

If Y is a topological subspace of X and B is some finite index set, then a **finite open cover V_\bullet of Y** is a collection of open subsets $V_\beta \subseteq X$ satisfying

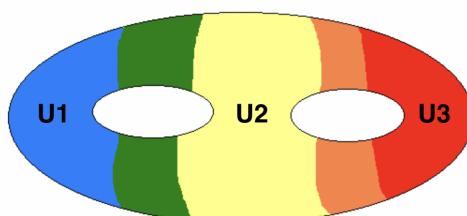
$$Y \subseteq \bigcup_{\beta \in B} V_\beta.$$

Definition 1.3.23. Let X be a topological space and let A be some finite index set. The **nerve $N(U_\bullet)$** of an open cover $\{U_\alpha : \alpha \in A\}$ of X is the simplicial complex whose i -simplices are given by all subsets $\sigma \subseteq A$ of cardinality $(i + 1)$, for which the intersection $Supp(\sigma) := \bigcap_{\alpha \in \sigma} U_\alpha$ is non-empty.

The intersection $Supp(\sigma)$ is called the support of σ .

The easiest way to understand this definition is through a visual example:

Example 1.3.24. Let X be a topological space and let $\{U_1, U_2, U_3\}$ be an open cover of it, that can be represented by the following picture:



The green color represents the overlap between U_1 and U_2 , whereas the orange color represents the overlap between U_2 and U_3 . The nerve $N(U_\bullet)$ of this open cover is the following simplicial complex:



By looking at this example, it is important to notice that a topological space may not have much in common with the geometric realization of the nerve of one of its open covers. This means that choosing an appropriate open cover is critical when trying to faithfully represent a topological space as a simplicial complex. However, we will see that under certain conditions, the geometric realization of the nerve of an open cover of a topological space X , is homotopy equivalent to X .

Example 1.3.25. Let (S, d) be a dataset. For $\epsilon > 0$, the **Čech complex** $C_\epsilon(S)$ is the nerve of the open cover $\{B_\epsilon(x) : x \in S\}$, where $B_\epsilon(x)$ denotes an open ball in \mathbb{R}^n of radius ϵ centered at x .

We will now introduce the Quillen's Fiber Lemma and its generalization, which is going to play a crucial part in the proof of The Nerve Theorem. However, before we proceed, we need to define what a fiber is:

Definition 1.3.26. Let K, L be simplicial complexes and let $f : K \rightarrow L$ be a simplicial map. For each simplex τ in L , the **fiber of f under τ** is the collection of simplices in K given by

$$\tau/f = \{\sigma \in K : f(\sigma) \leq \tau\}.$$

Lemma 1.3.27. (Quillen's Fiber Lemma) *Let K, L be simplicial complexes and let $f : K \rightarrow L$ be a simplicial map. If the fiber τ/f is contractible for every simplex τ in L , then the induced continuous map $|f| : |K| \rightarrow |L|$ admits a homotopy inverse $G : |L| \rightarrow |K|$. This implies that K and L are homotopy equivalent.*

Proof. Refer to Nanda (2021), Theorem 2.10. □

This lemma is extremely powerful since it allows us to conclude that two simplicial complexes are homotopy equivalent without having to find a homotopy inverse. Nonetheless, the main theorem's proof will require a generalization of this lemma.

Lemma 1.3.28. (*Generalization of Quillen's Fiber Lemma*) Let X, Y be locally contractible and Hausdorff topological spaces and let $f : X \rightarrow Y$ be a proper surjective continuous map. If for all $y \in Y$ we have that $f^{-1}(y)$ is contractible, then f induces a homotopy equivalence between X and Y .

Proof. Refer to Smale (1957), Main Theorem. \square

Equipped now with this lemma, we proceed to the main theorem of this subsection:

Theorem 1.3.29. (*The Nerve Theorem*) Let A be some finite index set and let $\{U_\alpha : \alpha \in A\}$ be a finite open cover of a topological space X . If for each simplex $\sigma \in N(U_\bullet)$, we have that $\text{Supp}(\sigma) \subseteq X$ is contractible, then $|N(U_\bullet)|$ is homotopy equivalent to X .

Proof. Let us define

$$X(U_\bullet) := \{(x, u) \in X \times |N(U_\bullet)| : \exists \sigma \in N(U_\bullet) \text{ with } x \in \text{Supp}(\sigma) \text{ and } u \in |\sigma|\}.$$

Since $X(U_\bullet)$ is a subset of $X \times |N(U_\bullet)|$, we can define the projection maps $\pi_X : X(U_\bullet) \rightarrow X$ as $\pi_X(x, u) = x$, and $\pi_U : X(U_\bullet) \rightarrow |N(U_\bullet)|$ as $\pi_U(x, u) = u$.

In order to use the generalization of Quillen's Fiber Lemma, we now need to show that for all $x \in X$ and $u \in |N(U_\bullet)|$, the fibers $\pi_X^{-1}(x)$ and $\pi_U^{-1}(u)$ are contractible. We proceed with these verifications.

1) π_X has contractible fibers: for any $x \in X$ we have that the fiber

$$\pi_X^{-1}(x) = \{(x, u) : u \in |N(U_\bullet)|, \exists \sigma \in N(U_\bullet) \text{ with } x \in \text{Supp}(\sigma) \text{ and } u \in |\sigma|\}$$

is homeomorphic to the set

$$\pi_U(\pi_X^{-1}(x)) = \{u \in |N(U_\bullet)| : \exists \sigma \in N(U_\bullet) \text{ with } x \in \text{Supp}(\sigma) \text{ and } u \in |\sigma|\}.$$

Notice that each $\sigma \in N(U_\bullet)$ satisfying $x \in \text{Supp}(\sigma)$ and $u \in |\sigma|$ for some $u \in \pi_U(\pi_X^{-1}(x))$, is a face of some highest dimensional simplex $\sigma_{max} \in N(U_\bullet)$ defined as

$$\sigma_{max}(x) := \{\alpha \in A : x \in U_\alpha\}.$$

Thus,

$$\pi_U(\pi_X^{-1}(x)) = |\sigma_{max}(x)|.$$

This implies $\pi_X^{-1}(x)$ is homeomorphic to $|\sigma_{max}(x)|$. Finally, notice that σ_{max} is the solid n -simplex for some integer n , which we already discussed is contractible in Example 1.3.21. Hence, π_X has contractible fibers.

Similarly, we prove the other case:

2) π_U has contractible fibers: for any $u \in |N(U_\bullet)|$ we have that the fiber

$$\pi_U^{-1}(u) = \{(x, u) : x \in X, \exists \sigma \in N(U_\bullet) \text{ with } x \in \text{Supp}(\sigma) \text{ and } u \in |\sigma|\}$$

is homeomorphic to the set

$$\pi_X(\pi_U^{-1}(u)) := \{x \in X : \exists \sigma \in N(U_\bullet) \text{ with } x \in \text{Supp}(\sigma) \text{ and } u \in |\sigma|\}.$$

Notice that there is a unique simplex $\sigma_u \in N(U_\bullet)$ which contains u in its geometric realization. Thus,

$$\pi_X(\pi_U^{-1}(u)) = \text{Supp}(\sigma_u).$$

This implies $\pi_U^{-1}(u)$ is homeomorphic to $\text{Supp}(\sigma_u)$, which is contractible by Theorem assumption. Hence, π_U has contractible fibers.

Since π_X, π_U have contractible fibers, we can apply the generalization of Quillen's Fiber Lemma. This yields the homotopy equivalences

$$X \simeq X(U_\bullet) \simeq |N(U_\bullet)|.$$

Therefore, X is homotopy equivalent to $|N(U_\bullet)|$.

□

Since we are going to be dealing with datasets, the following corollary is critical:

Corollary 1.3.30. *Let $M \subseteq \mathbb{R}^n$ be a finite set of points. For each radius $\epsilon > 0$, the union $M^{+\epsilon} \subseteq \mathbb{R}^n$ of radius ϵ Euclidean balls around the points of M is homotopy equivalent to the geometric realization of the Čech complex $C_\epsilon(M)$.*

Proof. We have that

$$M^{+\epsilon} = \bigcup_{x \in M} B_\epsilon(x).$$

Notice that the set $\{B_\epsilon(x) : x \in M\}$ is a finite open cover of M . Since the non-empty intersection of balls in this cover are convex subsets of \mathbb{R}^n , they are contractible. Recall now that, by definition, the Čech complex is the nerve of this open cover. Hence, by applying The Nerve Theorem, we obtain that $M^{+\epsilon}$ is homotopy equivalent to $|C_\epsilon(M)|$. □

This corollary is the reason why we have defined the Čech filtration, even though it is more involved than the Vietoris-Rips one. Moreover, there does not exist a similar result for the Vietoris-Rips complex $VR_\epsilon(M)$. This is the reason why we stated that Čech filtrations are seen as more accurate in capturing the topological structure of a dataset.

The Nerve Theorem and its corollary are of the utmost importance in topological data analysis, allowing us to represent a dataset as a Čech complex that accurately reflects its topological structure.

It is time now to move on to the next subsection, where we are going to explain how to identify some interesting properties of simplicial complexes.

1.3.2 Towards persistent homology

The main tool of topological data analysis is called persistent homology. In this subsection we will explore a particular version of it called *simplicial* persistent homology. With it, we will extract meaningful insights from our football dataset. Before introducing simplicial persistent homology, we require an understanding of several concepts. With this in mind, we start with our first definition:

Definition 1.3.31. Let K be a simplicial complex and let K_0 be its set of vertices. An **orientation of K** is an injective function $o : K_0 \rightarrow \mathbb{N}$ which assigns to each vertex a unique natural number. A simplicial complex with an orientation is called an **oriented simplicial complex**.

The number we assign to each vertex is not important. What matters is the ordering induced by o . Thus, we have the following definition:

Definition 1.3.32. Let K be a simplicial complex, $K_0 = \{v_0, v_1, \dots, v_n\}$ its set of vertices and $o : K_0 \rightarrow \mathbb{N}$ an orientation of K . We call $(w_0, w_1, \dots, w_n) \in K$ an **oriented simplex of K** if

$$o(w_0) < o(w_1) < \dots < o(w_n)$$

and each w_j is some distinct vertex $v_j \in K_0$.

In order to define homology, we first need to discuss some other notions that will be part of its definition. The following definitions and examples will be crucial for this purpose.

Definition 1.3.33. Let K be an oriented simplicial complex and let $\sigma = (v_0, v_1, \dots, v_n)$ be an oriented n -simplex in K . We define, for each $i \in \{0, 1, \dots, n\}$, the **i-th face** of σ

as the $(n - 1)$ -dimensional oriented simplex

$$\sigma_{-i} := (v_0, v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_n),$$

meaning the i -th coordinate is removed.

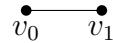
Definition 1.3.34. Let σ be an n -dimensional oriented simplex. The **algebraic boundary of σ** is defined as

$$\partial_n \sigma := \sum_{i=0}^n (-1)^i \sigma_{-i}.$$

By convention, ∂_0 of every vertex is 0 given that 0-simplices do not have lower-dimensional faces.

This concept can be easily understood with the following visual example:

Example 1.3.35. Consider the following 1-dimensional oriented simplex σ where $o(v_0) < o(v_1)$:



We compute $\partial_1(\sigma)$ as

$$\begin{aligned} \partial_1(\sigma) &= (-1)^0 \sigma_0 + (-1)^1 \sigma_1 \\ &= v_1 - v_0. \end{aligned}$$

Although this definition is not complicated, it is natural to ask what this formal sum of simplices means, and why it is important. The answers to these questions lie in the fact that we would like to have the powerful machinery of linear algebra at our disposal to use. In order to have these tools, we construct vector spaces out of simplicial complexes. In a way, this is the moment of this subsection where topology blends with algebra. The following definitions make this merge formal:

Definition 1.3.36. Let K be an oriented simplicial complex. We define, for each dimension $k \geq 0$, the **k -th chain group of K** as the vector space $C_k(K)$ over \mathbb{R} generated by a basis consisting of its k -dimensional simplices. Each element $\gamma \in C_k(K)$ is called a **k -chain** of K .

Naturally, every k -chain γ of K can be uniquely written as linear combination of the elements of the basis of $C_k(K)$ with coefficients in \mathbb{R} . As it is usual, we now define the respective linear maps between these vector spaces.

Definition 1.3.37. Let $C_k(K), C_{k-1}(K)$ be the k -th and $(k-1)$ -th chain groups of the oriented simplicial complex K over \mathbb{R} . We define the **k -th boundary operator of K** as the \mathbb{R} -linear map $\partial_k^K : C_k(K) \rightarrow C_{k-1}(K)$ that sends each basis k -chain σ to the $(k-1)$ -chain

$$\partial_k^K(\sigma) = \sum_{i=0}^k (-1)^i \sigma_{-i}.$$

With this definition, it becomes clear why we defined the algebraic boundary of an oriented simplex. Moreover, the ± 1 are now identified as elements of \mathbb{R} .

We now notice the following proposition:

Proposition 1.3.38. For every dimension $k \geq 0$, we have that

$$\partial_k^K \circ \partial_{k+1}^K : C_{k+1}(K) \rightarrow C_{k-1}(K)$$

is the zero map.

Proof. Refer to Nanda (2021), Proposition 3.8. □

Notice that this proposition is implying that the image of ∂_{k+1}^K is inside the kernel of ∂_k^K . Thus, we can visualize these vector spaces along with their respective linear maps as:

$$\dots \xrightarrow{\partial_{k+1}^K} C_k(K) \xrightarrow{\partial_k^K} C_{k-1} \xrightarrow{\partial_{k-1}^K} \dots \xrightarrow{\partial_2^K} C_1 \xrightarrow{\partial_1^K} C_0 \xrightarrow{\partial_0^K} 0$$

satisfying $\partial_j^K \circ \partial_{j+1}^K = 0$ for all $j \geq 0$. Due to the importance of this structure in this mathematical area, we have the following definition:

Definition 1.3.39. Let K be an oriented simplicial complex. A **simplicial chain complex** $(C_\bullet(K), \partial_\bullet^K)$ over \mathbb{R} is a collection of k -th chain groups $\{C_k(K) : k \geq 0\}$ along with the k -th boundary operators $\partial_k^K : C_k(K) \rightarrow C_{k-1}(K)$ that satisfy $\partial_k \circ \partial_{k+1} = 0$ for every integer $k \geq 0$.

We are now in a position to define homology:

Definition 1.3.40. Let $(C_\bullet(K), \partial_\bullet^K)$ be a simplicial chain complex over \mathbb{R} . We define, for each dimension $k \geq 0$, the **k -th simplicial homology group** of $(C_\bullet(K), \partial_\bullet^K)$ as the quotient vector space

$$H_k(K) := \ker(\partial_k^K) / \text{im}(\partial_{k+1}^K)$$

where \ker and im means the kernel and the image of a map, respectively.

We refer to $\ker(\partial_k^K)$ as the subspace of **k -cycles**, and to $\text{im}(\partial_{k+1}^K)$ as the subspace of **k -boundaries**. Thus, an ordinary phrase in algebraic topology is “homology is cycles modulo boundaries”.

Given that a finite vector space is completely determined by its dimension, we can identify each simplicial homology group by its dimension.

Example 1.3.41. Let K be an oriented simplicial complex which contains a unique 0-simplex, which we called the solid 0-simplex $\Delta(0)$. Constructing its simplicial chain complex over \mathbb{R} , we see that it is equal to

$$\dots \xrightarrow{\partial_3^K} 0 \xrightarrow{\partial_2^K} 0 \xrightarrow{\partial_1^K} \mathbb{R} \xrightarrow{\partial_0^K} 0.$$

We notice that the unique non-trivial kernel is $\ker(\partial_0^K)$ which has dimension 1. Therefore, the dimension of its simplicial homology groups are:

- $\dim(H_k(K)) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{if } k > 0 \end{cases}.$

Although these quotient vector spaces seem rather abstract, they carry a simple visual meaning. The 0-th, 1-st and 2-nd simplicial homology groups can be seen as representing the number of connected components, the number of “holes”, and the number of “voids” of a simplicial complex, respectively. For $k > 2$, the k -th simplicial homology groups are not so simple to visualize. Hence, if we are interested in computing low dimensional simplicial homology groups we can use these notions.

Example 1.3.42. Consider the following picture:

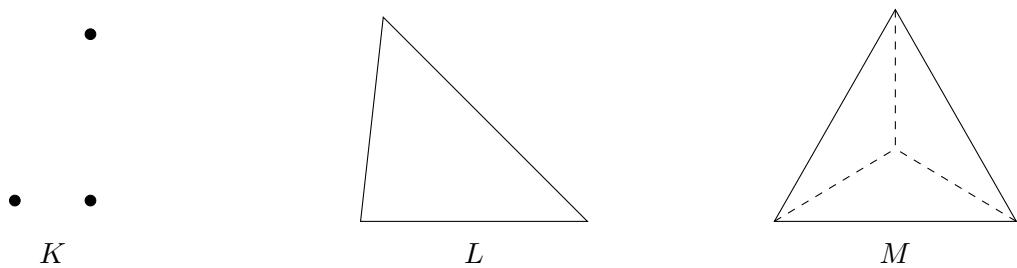


Figure 1.3: Three simplicial complexes: three points, a hollow triangle, and a hollow tetrahedron.

We now compute the dimensions of the k -th simplicial homology groups of K , L and M :

- $\dim(H_k(K)) = \begin{cases} 3 & \text{if } k = 0 \\ 0 & \text{if } k > 0 \end{cases}.$

$$\bullet \dim(H_k(L)) = \begin{cases} 1 & \text{if } k = 0 \\ 1 & \text{if } k = 1 \\ 0 & \text{if } k > 1. \end{cases}$$

$$\bullet \dim(H_k(M)) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{if } k = 1 \\ 1 & \text{if } k = 2 \\ 0 & \text{if } k > 2. \end{cases}$$

Homology is essential for understanding simplicial complexes. However, we are not only interested in the static version of topological features. We are also interested in how the structure of our objects evolve over some parameter. With this in mind, we continue with our final three definitions of this section:

Definition 1.3.43. Let K and L be oriented simplicial complexes. If $K \subseteq L$, the inclusion map $\iota : K \rightarrow L$ induces a map on their homology groups $\iota_k : H_k(K) \rightarrow H_k(L)$ defined by $z + im(\partial_{k+1}^K) \mapsto z + im(\partial_{k+1}^L)$, for each dimension $k \geq 0$.

Definition 1.3.44. A **persistence module** \mathcal{F} over \mathbb{R} indexed by the real numbers is a collection of \mathbb{R} -vector spaces $\{V_t : t \in \mathbb{R}\}$ along with linear maps $\phi_s^t : V_s \rightarrow V_t$ for all $s \leq t$, which satisfies the following:

- Identity: for any $t \in \mathbb{R}$, ϕ_t^t is the identity map on V_t .
- Composition: for any $r \leq s \leq t$, $\phi_s^t \circ \phi_r^s = \phi_r^t$.

\mathcal{F} can be visualized as follows:

$$\dots \rightarrow V_{t_0} \xrightarrow{\phi_{t_0}^{t_1}} V_{t_1} \xrightarrow{\phi_{t_1}^{t_2}} V_{t_2} \xrightarrow{\phi_{t_2}^{t_3}} \dots$$

where $\dots \leq t_0 \leq t_1 \leq t_2 \leq \dots$

Finally, we define the main tool of topological data analysis:

Definition 1.3.45. Let $\{K_t : t \in \mathbb{R}\}$ be an increasing family of simplicial complexes. The collection of \mathbb{R} -vector spaces $\{H_k(K_t) : t \in \mathbb{R}\}$ along with the induced maps on homology from inclusion is a persistent module, which we call **k -th simplicial persistent homology**.

Example 1.3.46. Let (S, d) be a dataset. Consider the Vietoris-Rips filtration (Definition 1.3.12) of S , denoted as $VR_\epsilon(S)$, which can be visualized as follows:

$$VR_{\epsilon_0}(S) \xrightarrow{\iota_{\epsilon_0}} VR_{\epsilon_1}(S) \xrightarrow{\iota_{\epsilon_1}} VR_{\epsilon_2}(S) \xrightarrow{\iota_{\epsilon_2}} \dots$$

where $\epsilon_0, \epsilon_1, \epsilon_2 \in \mathbb{R}_{\geq 0}$ such that $\epsilon_0 \leq \epsilon_1 \leq \epsilon_2$, and $\iota_{\epsilon_0}, \iota_{\epsilon_1}, \iota_{\epsilon_2}$ are the respective inclusion maps.

We compute the 0-th simplicial persistent homology of this filtration, which can be visualized as follows:

$$H_0(VR_{\epsilon_0}(S)) \xrightarrow{\phi_{\epsilon_0}^{\epsilon_1}} H_0(VR_{\epsilon_1}(S)) \xrightarrow{\phi_{\epsilon_1}^{\epsilon_2}} H_0(VR_{\epsilon_2}(S)) \xrightarrow{\phi_{\epsilon_2}^{\epsilon_3}} \dots$$

where each $\phi_{\epsilon_i}^{\epsilon_j}$ is the induced map from the inclusion $VR_{\epsilon_i}(S) \subseteq VR_{\epsilon_j}(S)$ for $i \leq j$.

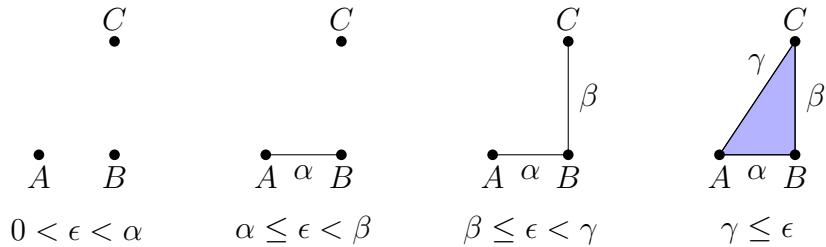
Although the example above is mathematically precise, a visual example will definitely help us enhance our knowledge. Before proceeding to it, we define a highly useful concept in topological data analysis:

Definition 1.3.47. Let (S, d) be a dataset and let $VR_\epsilon(S)$ be its Vietoris-Rips filtration. We define the $ST(VR_\epsilon(S))$ as the sum of the times at which previously disconnected components merge in the Vietoris-Rips filtration of S . These times are represented by the real values over which our parameter ϵ evolves.

Notice that given that the $ST(VR_\epsilon(S))$ refers to connected components, its computation is a consequence of H_0 . We continue with an example:

Example 1.3.48. We first compute, for all non-negative integers k , the k -th simplicial persistent homology of the Vietoris-Rips filtration from Example 1.3.14.

Vietoris-Rips filtration of S :



k -th simplicial persistent homology:

- $\dim(H_0(VR_\epsilon(S))) = \begin{cases} 3 & \text{if } 0 \leq \epsilon < \alpha \\ 2 & \text{if } \alpha \leq \epsilon < \beta \\ 1 & \text{if } \beta \leq \epsilon. \end{cases}$
- $\dim(H_k(VR_\epsilon(S))) = 0$ for all integers $k > 0$.

Computing the $ST(VR_\epsilon(S))$:

$$ST(VR_\epsilon(S)) = \alpha + \beta.$$

When discussing our football application, we will find simplicial persistent homology to be a formidable tool for obtaining important insights. In particular, the ST will be crucial for understanding how a new player influences a team.

Having explored all the topological data analysis concepts and tools we are interested in, it is time now to proceed to the next background section.

1.4 Category theory background

“To understand the question is very nearly to know the answer.”

Tom Leinster on category theory

Category theory, introduced by Eilenberg & MacLane (1945), is an area of mathematics that deals with mathematical structures and their relations. It acts as a bridge across almost every mathematical discipline, identifying common aspects, and providing a universal language. Moreover, category theory focuses on the connections between objects, instead of the elements inside them.

This section is, without a doubt, the densest and a principal reason why many people avoid delving into the details of UMAP. We will see that most of the ideas of this algorithm are written in categorical language, and that a vast number of categorical concepts are required to understand it. Hence, this module is indispensable for our purposes.

In this section, our examples are not going to be as detailed as in other parts of this thesis because their usually extensive proofs do not generally provide any new knowledge. In fact, if we understand the idea of an example, there is often a unique way to prove it. This is what Tom Leinster was trying to remark on his cited quote. Nonetheless, by no means this implies that this module is going to be superficial. We will carefully select the appropriate examples and add all the necessary comments to maximize our understanding.

This section is heavily influenced by the books written by Johnstone (2002), Mac Lane (2013), Leinster (2014), and Riehl (2017).

With this in mind, we proceed to our first categorical subsection.

1.4.1 Towards adjunction

In order to understand UMAP’s main theorem, it is essential to grasp the concept of adjunction in category theory, which can be described as a weaker notion of equivalence between some objects called “categories”. Thus, in this subsection, we will define all the theory needed to comprehend this definition.

We begin our journey with this first definition:

Definition 1.4.1. A category \mathcal{C} consists of:

- A collection of objects $\mathcal{O}_{\mathcal{C}}$.

- A collection of morphisms (also called arrows). Each morphism has objects in \mathcal{O}_C as domain and codomain. For $A, B \in \mathcal{O}_C$, $f : A \rightarrow B$ (also written as $A \xrightarrow{f} B$) means f is a morphism with domain A and codomain B . The notation of the collection of all morphisms from A to B in C is $hom_C(A, B)$.
- For $A, B, C \in \mathcal{O}_C$ and morphisms $f : A \rightarrow B$ and $g : B \rightarrow C$, there exists a morphism $gf := g \circ f : A \rightarrow C$. We can visualize this with the following commutative diagram:

$$\begin{array}{ccc} A & & \\ \downarrow f & \searrow gf & \\ B & \xrightarrow{g} & C \end{array}$$

- For every $A \in \mathcal{O}_C$, there exists an identity morphism $1_A : A \rightarrow A$.

Additionally, it satisfies the following axioms:

- Associativity: if $A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{h} D$, then $h(gf) = (hg)f$.
- Identity composition: if $A \xrightarrow{1_A} A \xrightarrow{f} B$, then $f1_A = f$. Similarly, if $A \xrightarrow{f} B \xrightarrow{1_B} B$, then $1_Bf = f$.

Unlike conventional maps, which define the transformation of individual elements of their respective domains, morphisms in categories work on a more abstract level. In category theory, the focus is not on specifying how a morphism acts on specific elements within its domain (actually, the elements within objects are not highly relevant). Instead, the emphasis lies on the relationships between morphisms. This is because objects per se are not of significant importance¹ in category theory. They are mostly useful to define morphisms, which are the heart of this area of mathematics.

It is important to remark that, thanks to its composition of morphisms property, we can picture categories through commutative diagrams.

Example 1.4.2. One of the simplest categories is the category of sets **SET**. Its objects are sets and its morphisms are maps between sets. Composition is given by composition of mappings. One can imagine many other examples of this kind: a mathematical structure with the appropriate maps. Topological spaces and continuous maps, vector spaces and linear maps are some of these.

Example 1.4.3. Let X be a topological space. There exists a category called **OPEN(X)** whose objects are the open sets of X and its morphisms are inclusion maps. Composition is given by composition of mappings.

¹In fact, the naming of specific categories after their objects is primarily due to historical conventions.

Example 1.4.4. The category Δ has finite ordered sets $[n] = \{0, 1, \dots, n\}$ as objects, and (non-strictly) order-preserving maps as morphisms. Composition is given by composition of mappings. It is also called the **simplex category**.

Example 1.4.5. The category of extended-pseudo-metric spaces \mathbf{EPMET} has extended-pseudo-metric spaces as objects, and non-expansive maps as morphisms. Composition is given by composition of mappings.

Unconventional example 1.4.6. Any single object A with morphisms being elements of a monoid M (a set with an associative binary operation and an identity element) constitutes a category. Each monoid element represents an endomorphism of the unique object, with composition given by multiplication. The identity morphism is represented by the identity element of M . Notice that these morphisms are not maps in the ordinary sense, but they still satisfy the category axioms.

Let us now delve into definitions and examples related to categories:

Definition 1.4.7. Let \mathcal{C} be a category and $A, B \in \mathcal{O}_{\mathcal{C}}$. A morphism $f : A \rightarrow B$ in \mathcal{C} is an **isomorphism** if there exists a morphism $g : B \rightarrow A$ in \mathcal{C} , such that $gf = 1_A$ and $fg = 1_B$.

Definition 1.4.8. A **subcategory** \mathcal{D} of a category \mathcal{C} is a collection of some objects and morphisms of \mathcal{C} , such that the subcategory \mathcal{D} contains the domain and codomain of any morphism in \mathcal{D} , the identity morphism 1_A for every $A \in \mathcal{O}_{\mathcal{D}}$, and the composite of any composable pair of morphisms in \mathcal{D} .

Example 1.4.9. The category of finite sets \mathbf{FINSET} is a subcategory of the category of sets \mathbf{SET} .

Example 1.4.10. The category of finite extended-pseudo-metric spaces $\mathbf{FINEPMET}$, is a subcategory of \mathbf{EPMET} . In this context, “finite” means that the underlying set X of each extended-pseudo-metric space (X, d) is finite.

Definition 1.4.11. Let \mathcal{D} be a subcategory of \mathcal{C} . We say \mathcal{D} is a **full subcategory of \mathcal{C}** if for any $A, A' \in \mathcal{O}_{\mathcal{D}}$, every morphism in $f : A \rightarrow A'$ in \mathcal{C} is also in \mathcal{D} .

Example 1.4.12. \mathbf{FINSET} is a full subcategory of \mathbf{SET} .

Definition 1.4.13. Let \mathcal{C}, \mathcal{D} be categories. The **product category** $\mathcal{C} \times \mathcal{D}$ have pairs of objects (A, B) with $A \in \mathcal{O}_{\mathcal{C}}$ and $B \in \mathcal{O}_{\mathcal{D}}$ as objects, and pairs of morphisms (f, g) where $f : A \rightarrow A'$ in \mathcal{C} and $g : B \rightarrow B'$ in \mathcal{D} . Composition is given by component-wise composition of mappings. Moreover, identity morphisms are given by $1_{A,B} := (1_A, 1_B)$.

Definition 1.4.14. Let \mathcal{C} be a category. The **opposite category** \mathcal{C}^{op} is a category with the same objects of \mathcal{C} but with every morphism reversed: for each morphism $f : A \rightarrow B$ in \mathcal{C} , $f^{op} : B \rightarrow A$ is a morphism in \mathcal{C}^{op} . The composite $f^{op}g^{op} := (gf)^{op}$ is defined in \mathcal{C}^{op} when the composite gf is defined in \mathcal{C} . For every object A , the morphism 1_A^{op} is its identity in \mathcal{C}^{op} . It is important to note that $(\mathcal{C}^{op})^{op} = \mathcal{C}$.

When facing this definition, a natural question may arise: “What happens when a morphism is not injective, and how do we define the opposite morphism?”. However, this is the wrong way of looking at morphisms. This is because objects and its elements are of little importance in this discipline, even though in some cases we will need to make reference to these elements. The right perspective is to consider a morphism just as an abstract arrow between abstract objects. Thus, an opposite morphism essentially involves reversing the direction of the arrow.

The notion of **duality** is of significant relevance in category theory. This term can be summarized in the following assertion: for every statement S that is true for a category \mathcal{C} , there exists a **dual** statement S^{op} that is true for its opposite category \mathcal{C}^{op} .

One important remark is that any proposition that states “For all categories $\mathcal{C}...$ ” also automatically applies to the opposite categories.

When initially learning mathematics, it is common to define objects of a particular kind and maps between them. However, we rarely see cases of maps between objects of different types. We will now talk about these maps by introducing two of the main concepts in category theory: functors and contravariant functors.

Definition 1.4.15. Let \mathcal{C}, \mathcal{D} be categories. A **functor** (also called covariant functor) $F : \mathcal{C} \rightarrow \mathcal{D}$ consists of:

- An object $F(A) \in \mathcal{D}$ for each object $A \in \mathcal{O}_{\mathcal{C}}$.
- A morphism $F(f) : F(A) \rightarrow F(A')$ in \mathcal{D} for each morphism $f : A \rightarrow A'$ in \mathcal{C} , so that the domain and codomain of $F(f)$ are equal to F applied to the domain or codomain of f .

Additionally, it satisfies the following functoriality axioms:

- For each $A \in \mathcal{O}_{\mathcal{C}}$, $F(1_A) = 1_{F(A)}$.
- For any composable pair of morphisms f, g in \mathcal{C} , $F(g)F(f) = F(gf)$.

Since categories can be thought as commutative diagrams, the functoriality of composition axiom highlights what could be considered a crucial attribute of functors: the preservation of commutative diagrams.

Example 1.4.16. Let \mathcal{C} be a category. The **identity functor** $\mathbf{1}_{\mathcal{C}}$ is the functor that maps each object $A \in \mathcal{O}_{\mathcal{C}}$ to itself, and each morphism $f : A \rightarrow A'$ in \mathcal{C} to itself.

Example 1.4.17. The **forgetful functor** is a term used to describe a functor that discards certain structure of its domain category. For instance, let \mathbf{GRP} be the category whose objects are all groups and its morphisms are group homomorphisms. The functor $F : \mathbf{GRP} \rightarrow \mathbf{SET}$, which transforms a group into its underlying set and a group homomorphism into its underlying set function, is a forgetful functor.

Example 1.4.18. The functor $F : \mathbf{SET} \rightarrow \mathbf{GRP}$ which maps each $A \in \mathcal{O}_{\mathbf{SET}}$ to its free group over A and each morphism $f : A \rightarrow A'$ in \mathbf{SET} to its induced homomorphism $F(f) : F(A) \rightarrow F(A')$ in \mathbf{GRP} , is called a **free functor**. Notice that this functor is well-defined since any map between sets gives rise to a homomorphism of the corresponding free groups.

Definition 1.4.19. Let \mathcal{C}, \mathcal{D} be categories. A **contravariant functor** F from \mathcal{C} to \mathcal{D} is a functor $F : \mathcal{C}^{\text{op}} \rightarrow \mathcal{D}$. Diagrammatically, for objects $A, A' \in \mathcal{O}_{\mathcal{C}}$ and a morphism $f : A \rightarrow A'$, we have

$$\begin{array}{ccc} A & & F(A) \\ \downarrow f & \mapsto & \uparrow F(f) \\ A' & & F(A') \end{array}$$

Example 1.4.20. The contravariant power set functor F from \mathbf{SET} to \mathbf{SET} is defined as the functor $F : \mathbf{SET}^{\text{op}} \rightarrow \mathbf{SET}$ that sends each object $A \in \mathcal{O}_{\mathbf{SET}}$ to its power set $P(A)$, and each morphism $f : A \rightarrow A'$ in \mathbf{SET} to $P(f) : P(A') \rightarrow P(A)$ which sends $B' \subseteq A'$ to $f^{-1}(B') \subseteq A$.

In order to introduce a category whose objects are (small) categories themselves, we need the following definition to address set-theoretic concerns (Russell's paradox):

Definition 1.4.21. Let \mathcal{C}, \mathcal{D} be categories. We say \mathcal{C} is **small** if the collection of objects and morphisms are sets. We say \mathcal{D} is **locally small** if for any pair of objects $A, B \in \mathcal{O}_{\mathcal{D}}$, $\text{hom}_{\mathcal{D}}(A, B)$ is a set.

Example 1.4.22. The category from Unconventional Example 1.4.6 constitutes a small category.

Example 1.4.23. SET is a locally small category. However, SET is not a small category because there is no set of all sets.

It is time now to present a key category in this discipline.

Example 1.4.24. The category with small categories as objects and functors as morphisms is called **CAT** . Composition is given by composition of functors. CAT is a locally small category but not a small one.

As it is usual in mathematics, we want to understand when two categories are “the same”, even though they may be labeled differently. So, we proceed with the following definition:

Definition 1.4.25. Let \mathcal{C}, \mathcal{D} be categories. An **isomorphism of categories** is a functor $F : \mathcal{C} \rightarrow \mathcal{D}$ such that there exists a functor $G : \mathcal{D} \rightarrow \mathcal{C}$, that satisfies $GF = 1_{\mathcal{C}}$ and $FG = 1_{\mathcal{D}}$, where $1_{\mathcal{C}}, 1_{\mathcal{D}}$ are the identity functors of the respective categories. An isomorphism of categories induces a bijection between the objects and morphisms of \mathcal{C} and \mathcal{D} .

Although this definition seems standard, we will find that this notion of isomorphism is overly restrictive, limiting its practical utility. To notice this, we need to define the notion of equivalence of categories, which first requires the following definitions:

Definition 1.4.26. Let \mathcal{C}, \mathcal{D} be categories and let $F, G : \mathcal{C} \rightarrow \mathcal{D}$ be functors. A **natural transformation** $\alpha : F \Rightarrow G$ consists of a morphism $\alpha_A : F(A) \rightarrow G(A)$ for each object $A \in \mathcal{O}_{\mathcal{C}}$, so that, for any morphism $f : A \rightarrow A'$ in \mathcal{C} , the following diagram commutes

$$\begin{array}{ccc} F(A) & \xrightarrow{F(f)} & F(A') \\ \alpha_A \downarrow & & \downarrow \alpha_{A'} \\ G(A) & \xrightarrow{G(f)} & G(A') \end{array}$$

Example 1.4.27. Let $P : SET \rightarrow SET$ be the power set functor that sends each object $A \in \mathcal{O}_{SET}$ to its power set $P(A)$, and each morphism $f : A \rightarrow A'$ in SET to $P(f) : P(A) \rightarrow P(A')$ where $P(f)(B) = \{f(b) : b \in B\}$ for $B \subseteq A$. There exists a natural transformation $\eta : 1_{SET} \Rightarrow P$ whose component $\eta_A : A \rightarrow P(A)$ maps $a \in A$ to the singleton set $\{a\} \in P(A)$.

Definition 1.4.28. Let \mathcal{C}, \mathcal{D} be categories. There exists a category named $[\mathcal{C}, \mathcal{D}]$ whose objects are functors from \mathcal{C} to \mathcal{D} and whose morphisms are natural transformations between them. Composition is given by composition of natural transformations.

Definition 1.4.29. Let \mathcal{C}, \mathcal{D} be categories and $F, G : \mathcal{C} \rightarrow \mathcal{D}$ be functors. A **natural isomorphism** is a natural transformation $\alpha : F \Rightarrow G$ such that for each object $A \in \mathcal{O}_{\mathcal{C}}$, $\alpha_A : F(A) \rightarrow G(A)$ is an isomorphism in \mathcal{D} . Two functors F and G are naturally isomorphic, denoted as $F \cong G$, if there exists a natural isomorphism between them.

With these definitions at hand, we can now define an equivalence of categories:

Definition 1.4.30. Let \mathcal{C}, \mathcal{D} be categories. An **equivalence of categories** is a functor $F : \mathcal{C} \rightarrow \mathcal{D}$ such that there exists a functor $G : \mathcal{D} \rightarrow \mathcal{C}$, that satisfies $GF \cong 1_{\mathcal{C}}$ and $FG \cong 1_{\mathcal{D}}$. The notation $\mathcal{C} \simeq \mathcal{D}$ means there exists an equivalence between the categories or, in other words, that \mathcal{C} and \mathcal{D} are **equivalent**.

The reason why an isomorphism of categories is not the right notion but equivalence of categories is, lies in the fact that the former implies a bijection on objects and morphisms, whereas the latter just implies a bijection on morphisms. Recall that objects are not of significant importance in this discipline.

In order to recognize an equivalence of categories, and to understand why an equivalence of categories does not imply a bijection on objects, we require the following definitions and theorem:

Definition 1.4.31. A functor $F : \mathcal{C} \rightarrow \mathcal{D}$ is:

- **Faithful** if for each $A, B \in \mathcal{O}_{\mathcal{C}}$, the map $hom_{\mathcal{C}}(A, B) \rightarrow hom_{\mathcal{D}}(F(A), F(B))$ is injective. A faithful functor that is injective on objects is called an **embedding**.
- **Full** if for each $A, B \in \mathcal{O}_{\mathcal{C}}$, the map $hom_{\mathcal{C}}(A, B) \rightarrow hom_{\mathcal{D}}(F(A), F(B))$ is surjective.
- **Essentially surjective on objects** if for every object $D \in \mathcal{O}_{\mathcal{D}}$ there exists some object $C \in \mathcal{O}_{\mathcal{C}}$ such that D is isomorphic to $F(C)$.

Theorem 1.4.32. *A functor defining an equivalence of categories is full, faithful, and essentially surjective on objects. Assuming the axiom of choice, any functor with these properties defines an equivalence of categories.*

Proof. Refer to Riehl (2017), Theorem 1.5.9. □

By this theorem, we understand that equivalences of categories only need essential surjectivity on objects, instead of bijectivity, making them a better concept in this discipline.

Example 1.4.33. Let $MAT_{\mathbb{K}}$ be the category with natural numbers as objects, and matrices with elements in a field \mathbb{K} , whose dimensions are given by the objects, as morphisms. Composition is given by matrix multiplication. Let $VECT_{\mathbb{K}}^{fd}$ be the category of finite dimensional \mathbb{K} -vector spaces. The functor $F : MAT_{\mathbb{K}} \rightarrow VECT_{\mathbb{K}}^{fd}$ which sends a natural number n to the vector space with the standard basis $F(n) = V_n$, and a matrix M to its linear map $F(M) = T_M$ in the standard basis, constitutes an equivalence of categories, because it is full, faithful and essentially surjective.

Equivalence of categories is definitely an important concept in category theory. However, when discussing UMAP, we will require an even weaker notion that we now introduce:

Definition 1.4.34. Let \mathcal{C}, \mathcal{D} be categories. An **adjunction** consists of an opposing pair of functors $F : \mathcal{C} \rightarrow \mathcal{D}$, $G : \mathcal{D} \rightarrow \mathcal{C}$, together with natural transformations $\eta : 1_{\mathcal{C}} \Rightarrow GF$ and $\epsilon : FG \Rightarrow 1_{\mathcal{D}}$, such that for any objects $A \in \mathcal{O}_{\mathcal{C}}$ and $B \in \mathcal{O}_{\mathcal{D}}$ the following diagrams commute

$$\begin{array}{ccc} F(A) & & \\ \downarrow F(\eta_A) & \searrow 1_{F(A)} & \\ FGF(A) & \xrightarrow{\epsilon_{F(A)}} & F(A) \\ \\ G(B) & & \\ \downarrow \eta_{G(B)} & \searrow 1_{G(B)} & \\ GFG(B) & \xrightarrow{G(\epsilon_B)} & G(B) \end{array}$$

We say that F is **left adjoint** to G , and G is **right adjoint** to F , denoted as $F \dashv G$.

Notice that the conditions set on the natural transformations η and ϵ , make them act as some sort of inverses of each other, even though they are not composable. This can be highlighted by looking at the commutativity of the diagram in equation form:

$$\begin{aligned} 1_{F(A)} &= \epsilon_{F(A)} \circ F(\eta_A), \\ 1_{G(B)} &= G(\epsilon_B) \circ \eta_{G(B)}. \end{aligned}$$

We should think of an adjunction as a relaxed form of an equivalence of categories. While equivalences of categories require natural isomorphisms, adjunctions only need some particular natural transformations.

Example 1.4.35. Let $F : SET \rightarrow GRP$ and $G : GRP \rightarrow SET$ be the free and forgetful functors we discussed in Example 1.4.18. These two functors constitute an adjunction $F \dashv G$. This is a particular case of what is called a **free \dashv forgetful adjunction**.

To conclude this subsection, we should keep this phrase in mind: “an adjunction can be thought of as a weaker notion of equivalence between categories”.

1.4.2 Towards sheaves

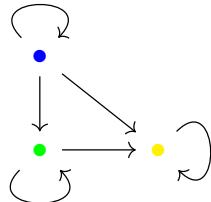
One of the main categories in UMAP has “sheaves” as its objects, so we will delve into the necessary concepts to be able to define what a sheaf is.

Informally, a diagram can be described as a directed graph. However, for our purposes, a precise categorical definition is necessary.

Definition 1.4.36. Let \mathcal{C} be a category. A **diagram \mathfrak{D} of shape \mathcal{I} in \mathcal{C}** is a functor $\mathfrak{D} : \mathcal{I} \rightarrow \mathcal{C}$ whose domain, the indexing category \mathcal{I} , is a small category.

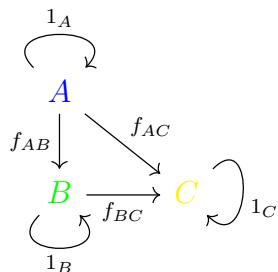
The indexing category \mathcal{I} plays the role of defining the shape of the diagram. This concept can be better understood with an example:

Example 1.4.37. Let \mathcal{I} be the category which has three objects (labeled by colors) and morphisms represented by:



Notice that we are implicitly assuming that the diagonal arrow is equal to the composition of the horizontal and vertical arrows.

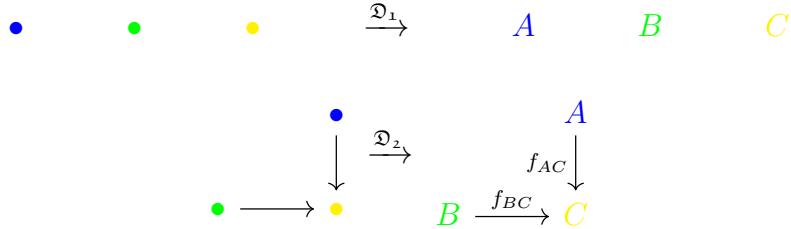
Let \mathcal{C} be a category. Let $A, B, C \in \mathcal{O}_{\mathcal{C}}$ (labeled by colors) and let $1_A : A \rightarrow A$, $1_B : B \rightarrow B$, $1_C : C \rightarrow C$, $f_{AB} : A \rightarrow B$, $f_{AC} : A \rightarrow C$, $f_{BC} : B \rightarrow C$ such that $f_{AC} = f_{BC}f_{AB}$, be morphisms in \mathcal{C} . We can define a diagram \mathfrak{D} of shape \mathcal{I} in \mathcal{C} as the functor $\mathfrak{D} : \mathcal{I} \rightarrow \mathcal{C}$ by:



Notice that $\mathfrak{D}(\bullet) = A$, $\mathfrak{D}(\bullet) = B$, $\mathfrak{D}(\bullet) = C$, and the arrows were mapped to the respective morphisms in \mathcal{C} .

In general, identity morphisms are not drawn for simplicity.

Example 1.4.38. Two different visual examples of diagrams:



Diagrams are essential in category theory, particularly for defining limits, a central concept in this area. In order to define limits, we first need to talk about constant functors and cones:

Definition 1.4.39. Let \mathcal{C}, \mathcal{I} be categories and let $A \in \mathcal{O}_{\mathcal{C}}$. The **constant functor at A** is the functor $\Delta A : \mathcal{I} \rightarrow \mathcal{C}$ that sends every object of \mathcal{I} to A and every morphism in \mathcal{I} to the identity morphism 1_A .

Definition 1.4.40. Let \mathcal{C} be a category and let \mathcal{I} be a small category. Let $\mathfrak{D} : \mathcal{I} \rightarrow \mathcal{C}$ be a diagram of shape \mathcal{I} in \mathcal{C} , and let $A \in \mathcal{O}_{\mathcal{C}}$. A **cone (A, λ) over \mathfrak{D}** is an object A together with a natural transformation $\lambda : \Delta A \Rightarrow \mathfrak{D}$ whose domain is the constant functor at A . The components $(\lambda_B : A \rightarrow \mathfrak{D}(B))_{B \in \mathcal{O}_{\mathcal{I}}}$ of the natural transformation are called the **legs** of the cone. For any morphism $f : B \rightarrow B'$ in \mathcal{I} , we have the following commutative diagram:

$$\begin{array}{ccc} & A & \\ \lambda_B \swarrow & & \searrow \lambda_{B'} \\ \mathfrak{D}(B) & \xrightarrow{\mathfrak{D}(f)} & \mathfrak{D}(B') \end{array}$$

Dually, a **cone (A, λ) under \mathfrak{D}** is an object A together with a natural transformation $\lambda : \mathfrak{D} \Rightarrow \Delta A$ whose codomain is the constant functor at A . For any morphism $f : B \rightarrow B'$ in \mathcal{I} , we have the following commutative diagram:

$$\begin{array}{ccc} \mathfrak{D}(B) & \xrightarrow{\mathfrak{D}(f)} & \mathfrak{D}(B') \\ \searrow \lambda_B & & \swarrow \lambda_{B'} \\ & A & \end{array}$$

Although the definitions of cones contain many concepts, the best way to understand them is by visualizing the commutative diagram they make, which is shaped like a cone. We are now able to explore the concepts of limits and colimits, which are central in this discipline.

Definition 1.4.41. Let \mathcal{C} be a category and let \mathcal{I} be a small category. Let $\mathfrak{D} : \mathcal{I} \rightarrow \mathcal{C}$ be a diagram of shape \mathcal{I} in \mathcal{C} , and let $A \in \mathcal{O}_{\mathcal{C}}$. A **limit** of a diagram \mathfrak{D} is a cone (A, λ) over \mathfrak{D} , such that given another cone (A', δ) over \mathfrak{D} , there is a unique morphism $f : A' \rightarrow A$ such that $\delta_B = \lambda_B f$ for each $B \in \mathcal{O}_{\mathcal{I}}$. Hence, we have the following commutative diagram:

$$\begin{array}{ccccc} & & A' & & \\ & \swarrow \delta_B & \downarrow f & \searrow \delta_{B'} & \\ \mathfrak{D}(B) & \xrightarrow{\mathfrak{D}(f)} & A & & \mathfrak{D}(B') \\ & \nwarrow \lambda_B & & \swarrow \lambda_{B'} & \\ & & A' & & \end{array}$$

Dually, a **colimit** of a diagram \mathfrak{D} is a cone (A, λ) under \mathfrak{D} , such that given another cone (A', δ) under \mathfrak{D} , there is a unique morphism $f : A \rightarrow A'$ such that $\delta_B = f \lambda_B$ for each $B \in \mathcal{I}$. Hence, we have the following commutative diagram:

$$\begin{array}{ccc} \mathfrak{D}(B) & \xrightarrow{\mathfrak{D}(f)} & \mathfrak{D}(B') \\ \swarrow \lambda_B & & \searrow \lambda_{B'} \\ & A & \\ \downarrow \delta_B & \downarrow f & \downarrow \delta_{B'} \\ & A' & \end{array}$$

Notice that we have denoted the unique morphism f with a dashed arrow. We will keep this notation throughout this thesis.

Limits and colimits can be thought of as the minimal type of diagrams that can be put on top of a given diagram. They are a way of constructing (minimal) new diagrams from existing ones. The word “minimal” refers to the uniqueness of morphisms property of their definitions. This uniqueness of morphisms is what makes a (co)limit to be the “closest” different diagram to a given diagram.

Example 1.4.42. A product is a limit of a diagram indexed by a discrete category. A **discrete** category is one whose only morphisms are the identity morphisms for each object. Whether a product exists depends on the category and on its particular objects. To enhance our intuition, consider the following: the product in the category SET is the usual Cartesian product. Let $A, B \in \mathcal{O}_{SET}$. A product of A and B is defined as $A \times B$ with the projections $\pi_A : A \times B \rightarrow A$, $\pi_B : A \times B \rightarrow B$ which satisfies that for any set $D \in \mathcal{O}_{SET}$ and every pair of morphisms $f_A : D \rightarrow A$ and $f_B : D \rightarrow B$, there exists a unique morphism $f : D \rightarrow A \times B$ that makes the diagram commutes. Diagrammatically,

this is

$$\begin{array}{ccc}
 & D & \\
 f_A \swarrow & \downarrow f & \searrow f_B \\
 A \times B & & \\
 \pi_A \nwarrow & & \searrow \pi_B \\
 A & & B
 \end{array}$$

Dually, a **coproduct** is a colimit of a diagram indexed by a discrete category. Similarly, the coproduct in the category SET is the disjoint union with the inclusion maps. Diagrammatically, this is

$$\begin{array}{ccc}
 A & & B \\
 i_A \searrow & & \swarrow i_B \\
 & A \sqcup B & \\
 f_A \swarrow & \downarrow f & \searrow f_B \\
 D & &
 \end{array}$$

Example 1.4.43. A **pullback** is a limit of a diagram of two morphisms with the same codomain. If given morphisms are $f : A \rightarrow C$ and $g : B \rightarrow C$, the pullback is typically denoted as $A \times_C B$, though the notation \times_C may be replaced by another depending on the particular case. As an example, in the category SET , intersection of subsets constitutes a pullback. Let $A, B, C, D \in \mathcal{O}_{SET}$ such that $A, B \subseteq C$. These inclusions can be expressed by the inclusion maps $i_A : A \rightarrow C$ and $i_B : B \rightarrow C$. A pullback of these three objects can be defined as $A \cap B$ with the inclusion maps $i_{A \cap B; A} : A \cap B \rightarrow A$, $i_{A \cap B; B} : A \cap B \rightarrow B$ which satisfies that for any set D and every pair of morphisms $f_A : D \rightarrow A$ and $f_B : D \rightarrow B$ there exists a unique morphism $f : D \rightarrow A \cap B$ that make the diagram commutes. Diagrammatically, this is

$$\begin{array}{ccccc}
 & D & & & \\
 & \searrow f & \nearrow f_B & & \\
 & A \cap B & & B & \\
 f_A \searrow & \downarrow i_{A \cap B; A} & \nearrow i_{A \cap B; B} & \downarrow i_B & \\
 A & \xrightarrow{i_A} & C & &
 \end{array}$$

Notice that, in a broad sense, limits tend to “create substructures” from objects. On the other hand, colimits tend to “create superstructures” by combining objects. This point of view will be useful when encountering these concepts later on.

Before defining sheaves, we first need to explore the concept of a cover and its related definitions.

Definition 1.4.44. Let \mathcal{C} be a category and let $U \in \mathcal{O}_{\mathcal{C}}$. A **cover** on U is a family of morphisms $\{\phi_i : U_i \rightarrow U\}_{i \in I}$ in \mathcal{C} , where I is an index set and $U_i \in \mathcal{O}_{\mathcal{C}}$ for all $i \in I$.

Notice that this categorical definition does not require, for example, that the union of the U_i be equal to U . This highlights the fact that category theory focuses on morphisms, instead of the underlying object elements. It is important to remark that, in this setting, not every morphism with a fixed target is a cover, but only those specifically identified as covers.

Definition 1.4.45. Let \mathcal{C} be a category. A **Grothendieck topology τ on \mathcal{C}** consists of a set $cov(\mathcal{C})$ called covering, which consists of covers for each object $U \in \mathcal{O}_{\mathcal{C}}$, satisfying the following properties

1. If $\psi : V \rightarrow U$ is an isomorphism, then $\{\psi\} \in cov(\mathcal{C})$.
2. If $\{\phi_i : U_i \rightarrow U\}_{i \in I} \in cov(\mathcal{C})$ and $\{\psi_{ij} : U_{ij} \rightarrow U_i\}_{j \in I_i} \in cov(\mathcal{C})$ for each U_i , then $\{\phi_i \circ \psi_{ij} : U_{ij} \rightarrow U\}_{(i,j) \in \prod_{i \in I} i \times I_i} \in cov(\mathcal{C})$.
3. If $\{\phi_i : U_i \rightarrow U\}_{i \in I} \in cov(\mathcal{C})$ and $\psi : V \rightarrow U$ is a morphism, then
 - The pullback $U_i \times_U V$ exists.
 - $\{\delta_i : U_i \times_U V \rightarrow V\}_{i \in I} \in cov(\mathcal{C})$.

This definition can be daunting when first met. The right way to approach this is by thinking in the topological definition of a cover. With this concept in mind we informally explain this definition: Property 1 implies that if two objects are the same, then each one covers the other. Property 2 states that if some objects U_i cover another object U , and each of these U_i are covered by other objects U_{ij} , then the objects U_{ij} (for all i, j) cover U . Property 3 tells us that, for some V related to U , intersecting V with the objects U_i that cover U , constitutes a cover of V .

Definition 1.4.46. A **site** (\mathcal{C}, τ) is a small category \mathcal{C} equipped with a Grothendieck topology τ .

Example 1.4.47. Let X be a topological space. An example of a Grothendieck topology on the category $OPEN(X)$ is given by specifying, for each $U \in \mathcal{O}_{OPEN(X)}$, a cover $\{\phi_i : U_i \rightarrow U\}_{i \in I}$ that satisfies $\bigcup_{i \in I} U_i = U$.

Example 1.4.48. Let \mathcal{C} be a category. The trivial Grothendieck topology on \mathcal{C} is given by specifying, for each $U \in \mathcal{O}_{\mathcal{C}}$, the trivial cover $\{\phi : U \rightarrow U\}$.

Example 1.4.49. Let \mathcal{C} and \mathcal{D} be two categories equipped with Grothendieck topologies. We can define the product Grothendieck topology on $\mathcal{C} \times \mathcal{D}$ in the ordinary way. This is for any objects $U^{\mathcal{C}} \in \mathcal{C}$, $U^{\mathcal{D}} \in \mathcal{D}$, a family of morphisms $\{(\phi_i^{\mathcal{C}}, \phi_j^{\mathcal{D}})\}_{(i,j) \in I \times J}$ is considered a cover of $(U^{\mathcal{C}}, U^{\mathcal{D}})$ if $\{\phi_i^{\mathcal{C}} : U_i^{\mathcal{C}} \rightarrow U^{\mathcal{C}}\}_{i \in I} \in cov(\mathcal{C})$ and $\{\phi_j^{\mathcal{D}} : U_j^{\mathcal{D}} \rightarrow U^{\mathcal{D}}\}_{j \in J} \in cov(\mathcal{D})$.

Before we can define a sheaf, it is crucial to first comprehend presheaves and matching families, since these concepts will be part of the sheaf definition.

Definition 1.4.50. Let \mathcal{C} be a category. A **presheaf on \mathcal{C}** is a functor $F : \mathcal{C}^{op} \rightarrow SET$.

Example 1.4.51. Let X be a topological space. Let $F : OPEN(X)^{op} \rightarrow SET$ be the functor that sends each open set $U \in \mathcal{O}_{OPEN(X)}$ to the set of continuous functions from U to \mathbb{C} , denoted as $C(U, \mathbb{C})$, and also sends each morphism $i_{U,V} : U \rightarrow V$ in $OPEN(X)$ to the restriction map $F(i_{U,V}) : C(V, \mathbb{C}) \rightarrow C(U, \mathbb{C})$. This functor constitutes a presheaf on $OPEN(X)$. Diagrammatically, this is

$$\begin{array}{ccc} U & & C(U, \mathbb{C}) \\ \downarrow i_{U,V} & \mapsto & \uparrow F(i_{U,V}) \\ V & & C(V, \mathbb{C}) \end{array}$$

Notice that each continuous function $f \in C(V, \mathbb{C})$ can be restricted to a smaller open set U , obtaining $f|_U : U \rightarrow \mathbb{C}$ which remains continuous.

Example 1.4.52. A **simplicial set** F is a presheaf on the simplex category Δ . We can think of $F([n])$ as the set whose elements are the n -simplices of the simplicial set F . For a morphism $f : [n - 1] \rightarrow [n]$ in Δ , we can consider $F(f) : F([n]) \rightarrow F([n - 1])$ to be some mapping of an n -simplex of F to one of its $(n - 1)$ -dimensional faces.

Simplicial sets can be thought of as an extension of simplicial complexes, allowing us to construct a wider² variety of topological objects.

Example 1.4.53. Let I be the interval³ $[0, 1] \subseteq \mathbb{R}$ with topology given by intervals of the form $[0, a)$ for $a \in (0, 1]$. A **fuzzy set**, in categorical terms, is a presheaf F on $OPEN(I)$ such that for any $f : [0, a) \rightarrow [0, b)$ in $OPEN(I)$, we have that $F(f) : F([0, b)) \rightarrow F([0, a))$ is injective.

²Simplicial sets can also represent “degenerate” simplices. We can notice this by the fact that Δ can also contain morphisms such as $f : [n] \rightarrow [n - 1]$ where some $i \in [n - 1]$ gets mapped twice. Then, $F(f) : F([n - 1]) \rightarrow F([n])$ is called a **degeneracy** map. The term “degenerate” comes from the fact that we are associating a lower-dimensional simplex A to a higher-dimensional one B , where one of the edges of B got collapsed to a vertex in A .

³In the UMAP paper, there is a minor inconsistency in the definition of I . It is defined as $I = (0, 1]$, but if that were the case, it would not be an element of the topology. We have corrected this typo.

To connect this categorical definition of fuzzy set with the classic one (Definition 1.1.1), we can think of $F([0, a))$ as the set of all elements with membership strength at least a . Hence, for $f : [0, a) \rightarrow [0, b)$ in $OPEN(I)$, we have that $F(f) : F([0, b)) \rightarrow F([0, a))$ maps injectively elements of membership strength at least b to elements of lower membership strength a . Notice that this injective map intuitively makes sense since elements with membership strength at least b also have membership strength at least a .

Definition 1.4.54. Let \mathcal{C} be a small category, F a presheaf on \mathcal{C} , (\mathcal{C}, τ) a site and, $S \in \tau(A)$ a cover of $A \in \mathcal{O}_{\mathcal{C}}$. A **matching family for S of elements in F** is a rule assigning to each $\phi_i : A_i \rightarrow A$ in S , an element $s_{\phi_i} \in F(A_i)$ such that whenever $f_i : B \rightarrow A_i$ and $f_j : B \rightarrow A_j$ satisfy $\phi_i \circ f_i = \phi_j \circ f_j$, we have $F(f_i)(s_{\phi_i}) = F(f_j)(s_{\phi_j})$.

This definition is definitely hard to grasp when first met. We will try to break it down to make it as easy to understand as possible. For that we will use an intuitive football theme example. Let us start by looking at the commutative diagram given by this definition:

$$\begin{array}{ccc} & B & \\ f_i \swarrow & & \searrow f_j \\ A_i & & A_j \\ \phi_i \searrow & & \swarrow \phi_j \\ & A & \end{array}$$

Then, apply the contravariant functor F :

$$\begin{array}{ccc} & F(B) & \\ F(f_i) \nearrow & & \nwarrow F(f_j) \\ F(A_i) & & F(A_j) \\ \downarrow F(\phi_i) & & \downarrow F(\phi_j) \\ F(A) & & \end{array}$$

Now that we have these diagrams to look at, we proceed with the football example:

Consider A to be some football field. Instead of looking at it as a whole, which can be complicated, we can break it down into smaller pieces which do not have to be disjoint, represented by the elements of $\{A_i\}_{i \in I}$. Each A_i is like a small section of the football field. Suppose that for each A_i we obtain information about the condition of the grass in that particular region. This grass information is contained in each $s_{\phi_i} \in F(A_i)$. Now, let B be some overlapping area between A_i and A_j for $i \neq j$. The condition $\phi_i \circ f_i = \phi_j \circ f_j$ guarantees that B is the same area of A , when looking at it through the lens of A_i and

A_j , respectively. Finally, the condition $F(f_i)(s_{\phi_i}) = F(f_j)(s_{\phi_j})$ implies that the grass on B should be the same that the one of A_i and A_j since B is their overlapping area. This is what is called a “compatibility” or “gluing” condition.

Lastly, we arrive at the main definition of this subsection:

Definition 1.4.55. Let (\mathcal{C}, τ) be a site with \mathcal{C} being a small category and τ a Grothendieck topology, and let F be a presheaf on \mathcal{C} . The presheaf F is a **sheaf** with respect to τ if for all $A \in \mathcal{O}_{\mathcal{C}}$, for each covering family $\{\phi_i : A_i \rightarrow A\}_{i \in I}$ in τ and a matching family $\{s_{\phi_i}\}_{i \in I}$ corresponding to it, there is a unique element $s \in F(A)$ such that $F(\phi_i)(s) = s_{\phi_i}$.

In order to understand this definition, we continue with our football theme explanation: The uniqueness of $s \in F(A)$ such that $F(\phi_i)(s) = s_{\phi_i}$, implies that there is a unique global grass information s that agrees with each local grass information s_{ϕ_i} in the region A_i . In other words, the information of the grass of the football field A , agrees with the local information given on each section A_i .

Example 1.4.56. The presheaf from Example 1.4.51 is a sheaf.

The following example is part of the UMAP paper and is one of the reasons why we had to define many of these tools.

Example 1.4.57. Following Example 1.4.53, the category $OPEN(I)$ can be endowed with a Grothendieck topology, as in Example 1.4.47. Presheaves on $OPEN(I)$ with morphisms given by natural transformations constitute a category. Moreover, a category of fuzzy sets can be built by restricting to the subcategory of presheaves that are fuzzy sets. These presheaves are (trivially) sheaves under the Grothendieck topology on $OPEN(I)$. The category **FUZZ** of fuzzy sets is then the full subcategory of sheaves on $OPEN(I)$ where every object is a fuzzy set.

Before ending this section, it is important to state the following theorem, whose implications will be useful when discussing UMAP.

Theorem 1.4.58. *Let \mathcal{C}, \mathcal{D} be categories and let $F : \mathcal{C} \rightarrow \mathcal{D}$, $G : \mathcal{D} \rightarrow \mathcal{C}$ be an adjunction where $F \dashv G$. Then F preserves colimits and G preserves limits.*

Proof. Refer to Leinster (2014), Theorem 6.3.1. □

This result shows that preserving colimits is a necessary condition for a functor to be a left adjoint. Similarly, preserving limits is a necessary condition for a functor to be a right adjoint.

To conclude this subsection, the important takeaways for our purposes are the concept of sheaf, that colimits “combine objects”, and that left adjoints preserve colimits.

1.4.3 Towards finite fuzzy simplicial sets

One of the two main categories in UMAP consists of finite⁴ fuzzy simplicial sets. In this subsection, we will discuss all the necessary concepts and theorems to fully understand these objects.

It is time now to define a simple yet important type of functor.

Definition 1.4.59. Let \mathcal{C} be a locally small category. For any object $A \in \mathcal{O}_{\mathcal{C}}$ we define the **functor represented by A** as $hom_{\mathcal{C}}(A, -) : \mathcal{C} \rightarrow SET$, such that for objects $B, B' \in \mathcal{O}_{\mathcal{C}}$ and a morphism $f : B \rightarrow B'$, we have

$$\begin{array}{ccc} B & & hom_{\mathcal{C}}(A, B) \\ \downarrow f & \mapsto & \downarrow hom_{\mathcal{C}}(A, f) \\ B' & & hom_{\mathcal{C}}(A, B') \end{array}$$

where $hom_{\mathcal{C}}(A, f) : hom_{\mathcal{C}}(A, B) \rightarrow hom_{\mathcal{C}}(A, B')$ maps $g \mapsto f \circ g$ for each morphism g in $hom_{\mathcal{C}}(A, B)$.

Dually, we define the **contravariant functor represented by A** as $hom_{\mathcal{C}}(-, A) : \mathcal{C}^{op} \rightarrow SET$, such that for objects $B, B' \in \mathcal{O}_{\mathcal{C}}$ and a morphism $f : B \rightarrow B'$, we have

$$\begin{array}{ccc} B & & hom_{\mathcal{C}}(B, A) \\ \downarrow f & \mapsto & \uparrow hom_{\mathcal{C}}(f, A) \\ B' & & hom_{\mathcal{C}}(B', A) \end{array}$$

where $hom_{\mathcal{C}}(f, A) : hom_{\mathcal{C}}(B', A) \rightarrow hom_{\mathcal{C}}(B, A)$ maps $g \mapsto g \circ f$ for each morphism g in $hom_{\mathcal{C}}(B', A)$. Notice that a contravariant functor represented by an object is a presheaf.

Definition 1.4.60. Let \mathcal{C} be a locally small category. A functor $F : \mathcal{C} \rightarrow SET$ is **representable** if there exists an object $A \in \mathcal{O}_{\mathcal{C}}$ and a natural isomorphism between F and the functor represented by A . We say that F is **represented by** the object A .

Example 1.4.61. The forgetful functor $F : GRP \rightarrow SET$ is represented by the group \mathbb{Z} . This is because for each $A \in \mathcal{O}_{GRP}$, $\alpha_A : F(A) \rightarrow hom_{GRP}(\mathbb{Z}, A)$ is an isomorphism

⁴In the UMAP paper, they are called bounded fuzzy simplicial sets. We will see that this change of name provides a better intuition.

in SET given by mapping $a \in F(A)$ to the unique group homomorphism that maps 1 to a . This is a bijection since every group homomorphism with domain \mathbb{Z} is determined by the image of 1 which in this case is equal to a .

Before introducing finite fuzzy simplicial sets, we need to discuss one of the most important theorems in mathematics and one of its corollaries. We will focus on their covariant versions, even though their contravariant ones are equally important.

Theorem 1.4.62. (*Yoneda lemma*): *Let \mathcal{C} be a locally small category and let $F : \mathcal{C} \rightarrow SET$ be any functor. Then, for any object $A \in \mathcal{O}_{\mathcal{C}}$, there is a bijection*

$$Nat(hom_{\mathcal{C}}(A, -), F) \cong F(A)$$

that maps a natural transformation $\alpha : hom_{\mathcal{C}}(A, -) \Rightarrow F$ to $\alpha_A(1_A) \in F(A)$. $Nat(hom_{\mathcal{C}}(A, -), F)$ means the set of natural transformations between $hom_{\mathcal{C}}(A, -)$ and F .

Proof. Let α be a natural transformation in $Nat(hom_{\mathcal{C}}(A, -), F)$. Since α is a natural transformation we know there exists $\alpha_B : hom_{\mathcal{C}}(A, B) \rightarrow F(B)$ for each $B \in \mathcal{O}_{\mathcal{C}}$, such that for any morphism $f : B \rightarrow B'$ we have the following commutative diagram:

$$\begin{array}{ccc} hom_{\mathcal{C}}(A, B) & \xrightarrow{hom_{\mathcal{C}}(A, f)} & hom_{\mathcal{C}}(A, B') \\ \alpha_B \downarrow & & \downarrow \alpha_{B'} \\ F(B) & \xrightarrow{F(f)} & F(B') \end{array}$$

In particular, there exists $\alpha_A : hom_{\mathcal{C}}(A, A) \rightarrow F(A)$, such that for any morphism $g : A \rightarrow B$ we have the following commutative diagram:

$$\begin{array}{ccc} hom_{\mathcal{C}}(A, A) & \xrightarrow{hom_{\mathcal{C}}(A, g)} & hom_{\mathcal{C}}(A, B) \\ \alpha_A \downarrow & & \downarrow \alpha_B \\ F(A) & \xrightarrow{F(g)} & F(B) \end{array}$$

Notice that $1_A \in hom_{\mathcal{C}}(A, A)$ since \mathcal{C} is a category. This commutative diagram implies

$$\begin{aligned} F(g) \circ \alpha_A(1_A) &= \alpha_B \circ hom_{\mathcal{C}}(A, g)(1_A) \\ &= \alpha_B \circ g \circ 1_A && (\text{By definition of } hom_{\mathcal{C}}(A, g)) \\ &= \alpha_B(g). \end{aligned}$$

Now, notice that $\alpha_A(1_A) = X$ for some $X \in F(A)$. Thus, we obtain

$$F(g)(X) = \alpha_B(g)$$

Note that α is uniquely determined by X . This is further illustrated by the fact that if we substitute B for any other B' and choose any other morphism $h : A \rightarrow B'$ we would obtain the following commutative diagram:

$$\begin{array}{ccc} \hom_{\mathcal{C}}(A, A) & \xrightarrow{\hom_{\mathcal{C}}(A, h)} & \hom_{\mathcal{C}}(A, B') \\ \alpha_A \downarrow & & \downarrow \alpha_{B'} \\ F(A) & \xrightarrow{F(h)} & F(B') \end{array}$$

which again implies

$$\begin{aligned} F(h) \circ \alpha_A(1_A) &= \alpha_{B'} \circ \hom_{\mathcal{C}}(A, h)(1_A) \\ &= \alpha_{B'} \circ h \circ 1_A \\ &= \alpha_{B'}(h), \end{aligned}$$

and then

$$F(h)(X) = \alpha_{B'}(h),$$

showing that X determines the value of $\alpha_{B'}(h)$ for any $h : A \rightarrow B'$. Hence, we can define a function $\phi : \text{Nat}(\hom_{\mathcal{C}}(A, -), F) \rightarrow F(A)$ such that $\alpha \mapsto \alpha_A(1_A)$.

In order to define an inverse of ϕ , which we call $\psi : F(A) \rightarrow \text{Nat}(\hom_{\mathcal{C}}(A, -), F)$, we need to construct, for each $Y \in F(A)$, a natural transformation $\psi(Y) : \hom_{\mathcal{C}}(A, -) \Rightarrow F$. For that we need to define, for each $M \in \mathcal{O}_{\mathcal{C}}$, a component $\psi(Y)_M : \hom_{\mathcal{C}}(A, M) \rightarrow F(M)$, ensuring that the naturality diagrams, as shown for $l : A \rightarrow M$ in \mathcal{C} , commute:

$$\begin{array}{ccc} \hom_{\mathcal{C}}(A, A) & \xrightarrow{\hom_{\mathcal{C}}(A, l)} & \hom_{\mathcal{C}}(A, M) \\ \psi(Y)_A \downarrow & & \downarrow \psi(Y)_M \\ F(A) & \xrightarrow{F(l)} & F(M) \end{array}$$

Following the diagram we have for $1_A \in \hom_{\mathcal{C}}(A, A)$:

$$\begin{aligned} F(l) \circ \psi(Y)_A(1_A) &= \psi(Y)_M \circ \hom_{\mathcal{C}}(A, l)(1_A) \\ &= \psi(Y)_M \circ l \circ a \\ &= \psi(Y)_M(l). \end{aligned}$$

Notice that, in order for ψ to be an inverse of ϕ , we require

$$\psi(Y)_A(1_A) = Y. \quad (1.1)$$

Hence, by naturality we need to define

$$\psi(Y)_M(l) := F(l)(Y). \quad (1.2)$$

Since M and l were arbitrary, we have completely determined $\psi(Y)$.

We now need to show that $\psi(Y)$ is a natural transformation. For this, we have to show, for a generic morphism $s : M \rightarrow N$ in \mathcal{C} , that the following diagram commutes:

$$\begin{array}{ccc} \hom_{\mathcal{C}}(A, M) & \xrightarrow{\hom_{\mathcal{C}}(A, s)} & \hom_{\mathcal{C}}(A, N) \\ \psi(Y)_M \downarrow & & \downarrow \psi(Y)_N \\ F(M) & \xrightarrow{F(s)} & F(N) \end{array}$$

Since $l \in \hom_{\mathcal{C}}(A, M)$, we can use it to test the commutativity of the diagram. Mapping it along left-bottom of the diagram we get $F(s) \circ \psi(Y)_M(l)$, which by (1.2) has to be

$$F(s) \circ \psi(Y)_M(l) = F(s) \circ F(l)(Y). \quad (1.3)$$

Mapping l along top-right we obtain $\psi(Y)_N \circ \hom_{\mathcal{C}}(A, s)(l)$, which by (1.2) has to be

$$\begin{aligned} \psi(Y)_N \circ \hom_{\mathcal{C}}(A, s)(l) &= F(sl)(Y) \\ &= F(s) \circ F(l)(Y). \quad (\text{By functoriality of } F. \quad (1.4)) \end{aligned}$$

Since (1.3) and (1.4) are equal, the naturality of $\psi(Y)$ is proven.

We finally proceed to verify that ψ is in fact the inverse of ϕ . We know that

$$\begin{aligned} \phi(\psi(Y)) &= \psi(Y)_A(1_A) && (\text{By definition of } \phi) \\ &= Y. && (\text{By (1.1)}) \end{aligned}$$

which proves the first direction of the inverse.

For the other direction, we know that $\psi(\phi(\alpha)) = \psi(\alpha_A(1_A))$. By (1.2), we have

$$\psi(\alpha_A(1_A))_M(l) = F(l)(\alpha_A(1_A)).$$

Since α is a natural transformation, the following diagram commutes:

$$\begin{array}{ccc} \hom_{\mathcal{C}}(A, A) & \xrightarrow{\hom_{\mathcal{C}}(A, l)} & \hom_{\mathcal{C}}(A, M) \\ \alpha_A \downarrow & & \downarrow \alpha_M \\ F(A) & \xrightarrow{F(l)} & F(M) \end{array}$$

which shows $F(l)(\alpha_A(1_A)) = \alpha_M(l)$. Thus,

$$\begin{aligned} \psi(\alpha_A(1_A))_M(l) &= F(l)(\alpha_A(1_A)) \\ &= \alpha_M(l), \end{aligned}$$

which shows $\psi(\phi(\alpha)) = \psi(\alpha_A(1_A)) = \alpha$, proving the second direction of the inverse.

Therefore, $\text{Nat}(\hom_{\mathcal{C}}(A, -), F) \cong F(A)$. \square

Corollary 1.4.63. (*The Yoneda embedding*). *Let \mathcal{C} be a locally small category. The functor $Y : \mathcal{C} \rightarrow [\mathcal{C}^{\text{op}}, \text{SET}]$ given by*

$$\begin{array}{ccc} A & & \hom_{\mathcal{C}}(-, A) \\ \downarrow f & \mapsto & \downarrow \hom_{\mathcal{C}}(A, f) \\ A' & & \hom_{\mathcal{C}}(-, A') \end{array}$$

defines a full and faithful embedding.

Proof. Refer to Riehl (2017), Corollary 2.2.8. \square

We will try to enlighten the essence of this corollary by what we term “the goalkeeper analogy”: an efficient way to assess a goalkeeper’s proficiency is by subjecting them to shots from players. In the categorical context, picture the goalkeeper as an object and the shots aimed at them as morphisms. Hence, the analogy can be expressed as follows: all information regarding an object A in a category \mathcal{C} is contained within $\hom_{\mathcal{C}}(-, A)$.

The Yoneda embedding shows that representable functors are indeed powerful since they contain all the information of the object they are represented by. By reexpressing objects in terms of morphisms pointing at them, we can often enhance our understanding of a category.

Since we understand the importance of representable functors, we are now able to define what fuzzy simplicial sets are, as described in the UMAP paper.

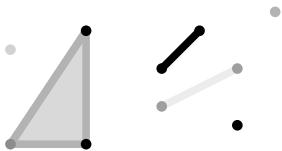
Example 1.4.64. The category of fuzzy simplicial sets \mathbf{SFUZZ} is the category with objects given by functors from Δ^{op} to $FUZZ$, and morphisms given by natural transformations. Alternatively, a fuzzy simplicial set can be viewed as a sheaf over $\Delta \times OPEN(I)$, where Δ is given the trivial Grothendieck topology, $OPEN(I)$ is given the Grothendieck topology from Example 1.4.57, and $\Delta \times OPEN(I)$ is given the product Grothendieck topology. We will use $\Delta_{\leq a}^n$ to denote the sheaf given by the representable functor of the object $([n], [0, a))$. Moreover, as might be expected, any fuzzy simplicial set X can be written as a combination of its fuzzy simplices $\Delta_{\leq a}^n$:

$$X \cong \text{colim}_{\Delta_{\leq a}^n \rightarrow X} \Delta_{\leq a}^n.$$

The notation on the right hand side should be read as merging the diagram of the fuzzy simplices $\Delta_{\leq a}^n$ that have a morphism pointing to X .

When explaining UMAP we will see that this sheafified version of a fuzzy simplicial set is deeply related to extended-pseudo-metric spaces.

We can think of a fuzzy simplicial set as a “shaded” simplicial set, where each simplex is colored using a gray scale which reflects its degree of membership. Moreover, the degree of membership of a simplex has to be greater or equal than the one of its faces, as seen in the fuzzy set definition in Example 1.4.53. We can visualize this concept with the following picture of a fuzzy simplicial set, where the darker the simplex, the higher its degree of membership:



When discussing UMAP, we will consider the subcategory of finite fuzzy simplicial sets $\mathbf{FIN-SFUZZ}$. A finite fuzzy simplicial set is a fuzzy simplicial set that is generated by finitely many vertices.

We now move on to the final background section.

1.5 Background on selected optimization methods

*“It is good to have people working from different fields;
it is nice to have a general culture in mathematics
and know about other fields.”*

Marguerite Frank

Real-life datasets often contain highly correlated variables. In general, it is in our best interest to delete these redundant features because having highly correlated variables distorts real distances between data points. This can be better illustrated with the following example:

Example 1.5.1. Let $D = \{(1, 1), (2, 1), (3, 2)\} \subseteq \mathbb{R}^2$ be a dataset where each data point coordinate represents a distinct non-highly correlated feature. The Euclidean distances between these data points are:

- $d((1, 1), (2, 1)) = 1 .$
- $d((1, 1), (3, 2)) = \sqrt{5} .$
- $d((2, 1), (3, 2)) = \sqrt{2} .$

Suppose we now add one redundant feature z such that $z = x$. Our dataset now becomes $D' = \{(1, 1, 1), (2, 1, 2), (3, 2, 3)\} \subseteq \mathbb{R}^3$. We compute their Euclidean distances again:

- $d((1, 1, 1), (2, 1, 2)) = \sqrt{2} .$
- $d((1, 1, 1), (3, 2, 3)) = 3 .$
- $d((2, 1, 2), (3, 2, 3)) = \sqrt{3} .$

Notice that the distances have changed, and this change has not been proportional, for example. Hence, any subsequent analysis that uses information about the pairwise distances of the data points will carry this inaccuracy.

We are now going to discuss some optimization methods that will help us determine which variables should be deleted from our football dataset. With this in mind, we start with some definitions:

Definition 1.5.2. Let A be an $n \times n$ matrix, where a_{ij} represents its element in row i and column j . The matrix A is called **diagonally dominant** if it satisfies, for all $1 \leq i \leq n$,

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|.$$

We call it **maximally diagonally dominant** if it is diagonally dominant and the inequality is strict for at least one row.

Example 1.5.3. The matrix $A = \begin{bmatrix} 4 & -2 & 2 \\ 2 & -5 & 3 \\ 1 & 0 & 3 \end{bmatrix}$ is diagonally dominant. It is also maximally diagonally dominant because $a_{33} > a_{31} + a_{32}$.

Definition 1.5.4. Let A, B be two real-valued $n \times n$ matrices. We define the **Frobenius inner product** of A and B as

$$\langle A, B \rangle_F := \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{ij}.$$

We will see in our football application that the question we are interested in answering is: “How do we reorder the rows and columns of a symmetric matrix with positive entries to make it maximally diagonally dominant?” The answer to this question is obtained by solving the following optimization problem:

Find $P \in \mathcal{P}_n$ that minimizes $\langle P \cdot A \cdot P^T, C \rangle_F$

where:

- \cdot denotes matrix multiplication.
- \mathcal{P}_n is the set of permutation matrices of size of $n \times n$.
- A is the matrix we want to sort.
- C is a cost matrix that penalises off-diagonal entries.

Notice that, since A and C are inputs, we start with some cost $\langle A, C \rangle_F$. In order to minimize this cost, we “modify” A by applying permutation matrices to it until we reach the desired precision.

Before introducing a solution to this problem, we need to understand the following definitions:

Definition 1.5.5. Let A be a non-negative real-valued $n \times n$ matrix. We call A **bistochastic** if each of its rows and columns sum 1.

Example 1.5.6. The matrix $A = \begin{bmatrix} 0.1 & 0.7 & 0.2 \\ 0 & 0.25 & 0.75 \\ 0.9 & 0.05 & 0.05 \end{bmatrix}$ is bistochastic.

Definition 1.5.7. The **Sinkhorn Algorithm** is a simple iterative method that takes a square matrix with non-negative entries and approximates it to a bistochastic one. It works by iteratively scaling a matrix's rows and columns until their sums are “close” to 1. This algorithm requires a precision hyperparameter to determine when to stop the iterations.

As noticed in Knight (2008), given an $n \times n$ matrix A with non-negative entries, the most useful interpretation of the Sinkhorn Algorithm is as a sequence of scaling diagonal matrices $(D_i, E_i)_{i=0}^{\infty}$ with D_0, E_0 being the $n \times n$ identity matrices, such that the $\lim_{m \rightarrow \infty} D_m \cdot A \cdot E_m$ is the desired bistochastic approximation of A .

Definition 1.5.8. Let P be an $n \times n$ bistochastic matrix. We define a convenient form of the **Shannon entropy of P** as

$$H(P) := - \sum_{i=1}^n \sum_{j=1}^n p_{ij} (\log(p_{ij}) - 1).$$

In our context, the Shannon entropy is going to be added as a regularization parameter to the function we want to minimize. Its addition will help the algorithm converge to better solutions by avoiding premature convergence.

We are now able to tackle our problem. In Nichols (2024), the solution for re-ordering a symmetric matrix with positive entries into a maximally diagonally dominant one is given by the two following algorithms:

The **Entropic Frank–Wolfe with Line-Search Algorithm** is an optimization method that works as follows: it takes as inputs a matrix A , a cost matrix C , an entropy regularization parameter ϵ , and the number of iterations n_{int} . Then,

Algorithm 1 Entropic Frank-Wolfe with Line-Search Algorithm (EFWLS)

```

1: procedure EFWLS( $A, C, \epsilon, n_{iter}$ )
2:    $n \leftarrow$  number of rows in  $A$ 
3:    $P_i \leftarrow I_n$                                  $\triangleright$  Set  $P_i$  to the  $n \times n$  identity matrix  $I_n$ 
4:   for  $i = 1$  to  $n_{iter}$  do
5:      $c_{P_i} \leftarrow \langle P_i \cdot A \cdot P_i^T, C \rangle_F$ 
6:      $\nabla \leftarrow$  compute gradient of  $c_{P_i}$  with respect to  $P_i$ 
7:      $Q_i \leftarrow$  find matrix  $Q$  that minimizes  $\langle Q, \nabla \rangle_F - \epsilon H(Q)$  using Sinkhorn
8:      $\eta_i \leftarrow$  find step size  $\eta \in [0, 1]$  that minimizes
9:        $\langle [\eta Q_i + (1 - \eta)P_i] \cdot A \cdot [\eta Q_i + (1 - \eta)P_i]^T, C \rangle_F$ 
10:     $P_i \leftarrow \eta_i Q_i + (1 - \eta_i)P_i$ 
11:   end for
12:   return  $P_i$ 

```

Let us unpack the most important characteristics of this algorithm. As previously stated, we want to find P that minimizes $\langle P \cdot A \cdot P^T, C \rangle_F$. Instead of minimizing this function, we use a method similar to Frank-Wolfe optimization where we find the gradient of $\langle P \cdot A \cdot P^T, C \rangle_F$, and optimize against the gradient with the addition of entropy, iteratively. The use of the Sinkhorn Algorithm is because it is known to be highly effective with solutions that are constrained to be bistochastic. The reason why we are restricting our solution to be bistochastic is because we want to find a matrix that is close to a permutation matrix, and permutation matrices are a subset of bistochastic matrices. If we were to restrict our solution to be directly a permutation matrix, we would not have a continuous optimization problem, which is why we need this bistochastic relaxation. Then, rather than choosing a fixed step size η or making an arbitrary guess, we find η via line-search that will help us converge more efficiently to our desired solution. Finally, we modify our P so that it gets closer to an optimal answer.

Ultimately, to solve the question of how to transform a positive symmetric matrix into a maximally diagonally dominant one, we use the following algorithm:

Let A be the matrix we want to sort, ϵ an entropy regularization parameter, and n_{int} the number of iterations of our algorithm. For our purposes, we define the **cost matrix** C to be a square matrix whose entries are defined as $c_{ij} = |i - j|^2$. Notice that it penalizes off-diagonal entries. Then,

Algorithm 2 Matrix Sorting Algorithm

```

1: procedure SORTMATRIX( $A, \epsilon, n_{iter}$ )
2:    $n \leftarrow$  number of rows in  $A$ 
3:    $C \leftarrow$  create  $n \times n$  cost matrix
4:    $P_e \leftarrow$  EFWLS( $A, C, \epsilon, n_{iter}$ )
5:    $P \leftarrow$  find nearest permutation matrix to  $P_e$ 
6:   return  $P \cdot A \cdot P^T$                                  $\triangleright$  Return sorted matrix

```

The nearest permutation matrix P is often effortless to find from a given bistochastic matrix P_e because it tends to be sufficiently close.

Remark 1.5.9. *Finding the nearest permutation matrix can also be done using the **Hungarian Algorithm** (Kuhn (1955)). However, since P_e is close enough to a permutation matrix P , it is less computationally expensive to do a binary threshold, which is what Algorithm 2 does.*

Having finished our background sections, it is time now to proceed to the next chapter of our thesis.

Dimensionality reduction algorithms

Dimensionality reduction algorithms are tools that transform high-dimensional data into a low-dimensional format while preserving its meaningful attributes.

There are many reasons to use dimensionality reduction algorithms, such as eliminating redundant features and reducing computation time. However, in this thesis, our principal goal is to enhance visualization which can only be achieved if datasets are represented in at most three dimensions.

The ability to visualize data is extremely useful since it can help us identify the right questions to ask, and also highlight points of interest that would remain hidden otherwise.

We will discuss three of the most significant algorithms: PCA, UMAP and Mapper. Each of them has its advantages and disadvantages, and the choice of the most appropriate method depends on the particular dataset we are working with.

2.1 PCA

Principal component analysis (PCA), first described by Pearson (1901) and subsequently by Hotelling (1933), is considered to be the first dimensionality reduction algorithm. In order to understand PCA, some background knowledge in linear algebra is needed. Since these needed topics are elementary, we will not spend much time on their details. Our explanation is heavily influenced by Shlens (2014).

This algorithm can be outlined as follows: a high-dimensional dataset, which is expressed as an $m \times n$ matrix, is given as input. PCA starts by computing the covariance matrix of this dataset. Then, it calculates the eigenvalues and eigenvectors of the covariance matrix, sorting these latter ones by their decreasing eigenvalues. Next, it chooses the first k eigenvectors, which are called principal components, and builds an eigenvectors matrix. Finally, it projects the original dataset into a k -dimensional representation, by multiplying the dataset with the eigenvectors matrix.

Before delving into PCA's mechanics, we begin with a football theme toy example that will enhance our intuition:

Foosball is a table game based on football. Its objective is to hit the ball with the players, which are attached to rods, in order to score as many goals as possible and to concede the least number of them. In our toy example, we are interested in understanding how the goalkeeper of a foosball team moves during a match. We have arbitrarily placed cameras A and B to capture, at different times, the position of one of the goalkeepers. We can visualize the situation with the following picture:

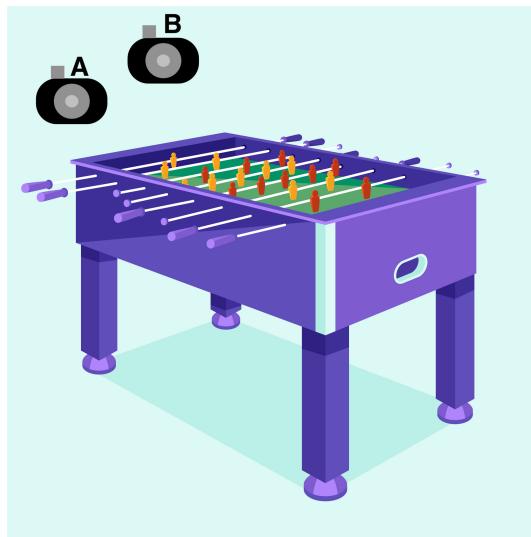


Figure 2.1: Two cameras recording the position of the yellow goalkeeper in a foosball game, image by Freepik (with modifications).

Each camera, with respect to its own point of view, outputs the x and y position of the yellow goalkeeper at different times t . Suppose each camera records n snapshots. This yields four vectors

$$\begin{aligned}x^A &= [x_{t_1}^A \ x_{t_2}^A \ \dots \ x_{t_n}^A], \\y^A &= [y_{t_1}^A \ y_{t_2}^A \ \dots \ y_{t_n}^A], \\x^B &= [x_{t_1}^B \ x_{t_2}^B \ \dots \ x_{t_n}^B], \\y^B &= [y_{t_1}^B \ y_{t_2}^B \ \dots \ y_{t_n}^B].\end{aligned}$$

A smart observer of the situation would realize that, since the goalkeeper is attached to a rod, it can only move in one direction. Thus, using four vectors to capture the yellow goalkeeper position definitely contains redundant information.

In this toy example, it is clear that we could understand the entire motion of the yellow goalkeeper considering only one direction. However, for more complicated datasets, finding out what are the most important directions can be a challenging task. The goal of PCA is to determine the crucial directions that contain the most information in our dataset.

It is time now to explore the assumptions and technicalities of PCA:

A dataset, represented as an $m \times n$ matrix, has snapshots of an object we are trying to understand as columns, and features of the object as rows. In our toy example, the first row

$$x^A = [x_{t_1}^A \ x_{t_2}^A \ \dots \ x_{t_n}^A]$$

represents the goalkeeper position given by the x -axis of the camera A , at n different times. On the other hand, the first column

$$\begin{bmatrix} x_{t_1}^A \\ y_{t_1}^A \\ x_{t_1}^B \\ y_{t_1}^B \end{bmatrix}$$

represents the goalkeeper position given by the two cameras, at time t_1 .

In order to understand what PCA does, we first need to define what the covariance matrix of a dataset is. We proceed with the required definitions.

Let Z be a dataset represented as an $m \times n$ matrix

$$Z = \begin{bmatrix} z_{t_1}^1 & z_{t_2}^1 & \dots & z_{t_n}^1 \\ z_{t_1}^2 & z_{t_2}^2 & \dots & z_{t_n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ z_{t_1}^m & z_{t_2}^m & \dots & z_{t_n}^m \end{bmatrix}.$$

We first center the data so that it has zero mean. This implies computing, for each i such that $1 \leq i \leq m$, the mean $M^i \in \mathbb{R}$ defined as

$$M^i := \frac{\sum_{j=1}^n z_{t_j}^i}{n}$$

and subtracting M^i from each entry in the row i .

We can now proceed with our explanation with the dataset X , defined as the $m \times n$ matrix

$$X = \begin{bmatrix} x_{t_1}^1 & x_{t_2}^1 & \dots & x_{t_n}^1 \\ x_{t_1}^2 & x_{t_2}^2 & \dots & x_{t_n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{t_1}^m & x_{t_2}^m & \dots & x_{t_n}^m \end{bmatrix} = \begin{bmatrix} z_{t_1}^1 - M^1 & z_{t_2}^1 - M^1 & \dots & z_{t_n}^1 - M^1 \\ z_{t_1}^2 - M^2 & z_{t_2}^2 - M^2 & \dots & z_{t_n}^2 - M^2 \\ \vdots & \vdots & \ddots & \vdots \\ z_{t_1}^m - M^m & z_{t_2}^m - M^m & \dots & z_{t_n}^m - M^m \end{bmatrix}$$

where each row has zero mean, meaning for any i such that $1 \leq i \leq m$ we have

$$\frac{\sum_{j=1}^n x_{t_j}^i}{n} = 0.$$

Working with this centered dataset will simplify upcoming computations.

We denote the i -th row of X by \mathbf{x}^i .

We define the variance of \mathbf{x}^i as

$$Var(\mathbf{x}^i) := \frac{\sum_{j=1}^n (x_{t_j}^i)^2}{n}.$$

Similarly, for any k such that $1 \leq k \leq m$, we define the covariance of \mathbf{x}^i and \mathbf{x}^k as

$$Cov(\mathbf{x}^i, \mathbf{x}^k) := \frac{\sum_{j=1}^n x_{t_j}^i x_{t_j}^k}{n}.$$

Notice that we can also express $Cov(\mathbf{x}^i, \mathbf{x}^k)$ as

$$Cov(\mathbf{x}^i, \mathbf{x}^k) = \frac{\mathbf{x}^i (\mathbf{x}^k)^T}{n}$$

where T is the transpose operator.

$Cov(\mathbf{x}^i, \mathbf{x}^k)$ quantifies how linearly correlated \mathbf{x}^i and \mathbf{x}^k are. The bigger its absolute value, the more correlated they are. The covariance of two features represents their redundancy. This is because most of the information contained in two highly correlated features is also contained in each of them separately.

Since X can also be expressed as

$$X = \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \\ \vdots \\ \mathbf{x}^m \end{bmatrix}$$

we define the covariance matrix of X as

$$C_X := \frac{XX^T}{n}.$$

Notice that the diagonal cells of C_X consist of the variance of the features, whereas the off-diagonal terms represent the covariance. Also, note that C_X is symmetric since

$$(XX^T)^T = (X^T)^T X^T = XX^T.$$

With these concepts in mind, we arrive at PCA's assumptions:

Assumption 1: large variances have important structure

Although this assumption highlights interesting structure in a multitude of datasets, it also fails to provide insights on many others. Thus, this assumption that makes the foundation of the algorithm, also constitutes its limitations. Moreover, it is implicitly implying that lower variances represent noise in our measurements.

It is important to notice that this assumption implies that the diagonal cells of C_X are crucial. If we were to transform our dataset X into a lower-dimensional one Y , we would like the covariance matrix of Y to be diagonal. This would mean that the off-diagonal cells are zero, implying that our new dataset features are less correlated, hence reducing the redundancy.

Assumption 2: linearity

A natural and implicit assumption that PCA makes is that the data was gathered using the standard basis. What PCA seeks to find is if there exists another basis, which is a linear combination of the standard, that can better express our dataset. We consider this re-expression to be satisfactory if it can reveal some structure, that was hidden before due to redundant and noisy information.

By assuming linearity, our problem simplifies to finding an appropriate change of basis matrix P , such that $PX = Y$ and $C_Y := \frac{YY^T}{n}$ is diagonal. The rows of P are called the **principal components** of P .

In order to simplify its problem even more, PCA makes one final assumption:

Assumption 3: the principal components are orthogonal

This final assumption makes it possible to solve our problem just by using linear algebra. We proceed with PCA's main proposition and its proof:

Proposition 2.1.1. (PCA) Let X be a dataset. There exists a orthonormal matrix P such that $PX = Y$ and C_Y is diagonal.

Proof. First, we rewrite C_Y as

$$\begin{aligned} C_Y &= \frac{YY^T}{n} \\ &= \frac{PX(PX)^T}{n} \\ &= \frac{PXX^TP^T}{n} \\ &= P\frac{XX^T}{n}P^T \\ &= PC_XP^T. \end{aligned}$$

Earlier on, we showed that C_X is symmetric. So, we can make use of fact that says that a symmetric matrix is diagonalized by a matrix of its orthonormal eigenvectors. In order to apply this, we construct P as a matrix where each row is an orthonormal eigenvector of $\frac{XX^T}{n}$. By making use of the stated fact, we obtain that $C_X = P^TDP$, where D is diagonal. Notice that, since P is orthonormal, we have that $P^{-1} = P^T$. We proceed to

check that P fulfill our demands

$$\begin{aligned} C_Y &= PC_X P^T \\ &= P(P^T D P)P^T \\ &= (PP^T)D(PP^T) \\ &= (PP^{-1})D(PP^{-1}) \\ &= D. \end{aligned}$$

Therefore, we have found some orthonormal matrix P that satisfies the proposition requirements. \square

Notice that if we sort the set of eigenvectors of C_X in decreasing order according to their eigenvalues, and construct P following this order, then the matrix C_Y will be diagonal with decreasing values. We can then define P_k as the matrix whose rows are the first k rows of P . By using P_k we can project X into k dimensions that maximize its variance.

In the implementation of PCA, k is a hyperparameter that must be defined, and it has to be an integer smaller than the number of dimensions of our dataset. The hope in performing this procedure is that the k dimensions will capture most of the variance of the dataset, indicating that the lower-dimensional representation can appropriately characterize our dataset.

Although PCA is an old algorithm, it does not make it unhelpful. In fact, since it only requires k to be set, it makes it a great starting point when facing a new dataset.

When discussing our football application, we will use PCA as a departure point. However, to get a deeper analysis of our dataset, we need to talk about one of the state-of-the-art dimensionality reduction algorithms. Thus, we proceed to our next section.

2.2 UMAP

Uniform Manifold Approximation and Projection (UMAP), introduced by Leland McInnes (McInnes et al. (2018)), is one of the most prominent dimensionality reduction algorithms. Unlike many algorithms typically developed through trial and error, UMAP was conceived using the author’s extensive mathematical knowledge.

In order to understand the essence of UMAP, it is necessary to have a solid background in several mathematical areas. Nonetheless, it is a challenging task to fully grasp UMAP’s mathematical ideas, given that its paper contains some inconsistencies and lacks precise definitions in certain parts. Although UMAP’s implementation is peer-reviewed, the mathematical concepts behind it have not received the same level of scrutiny. Additionally, much of UMAP’s categorical ideas are based on Spivak (2009), which is a preprint. While Spivak’s paper is enlightening, it could benefit from greater clarity in its exposition. Furthermore, we will notice that there is a clear disconnection between the mathematics that inspired UMAP and the actual algorithm.

UMAP can be summarized as follows: it begins by approximating the manifold from which the data was sampled. Then, it constructs a finite fuzzy simplicial set which captures the topological structure of the manifold. Next, UMAP initializes a lower-dimensional representation and constructs a finite fuzzy simplicial set that represents it. Finally, the algorithm iteratively optimizes the lower-dimensional representation to make both finite fuzzy simplicial sets as similar as possible.

Due to the nature of this thesis, our focus will be on the mathematical ideas of UMAP rather than on its implementation, though these two aspects will occasionally overlap. Our endeavor will be to explain the concepts, address as many inconsistencies as we can, and add mathematical rigor where necessary. With this in mind, we proceed with our explanation.

2.2.1 Introduction to UMAP

We start by considering UMAP’s first assumption:

Assumption 1: data was sampled from a manifold

To enhance our understanding, we will not start by explaining UMAP in abstract. The reason for this choice is that we believe that having a visual example to follow is crucial to comprehend what is going on, at least during the first part of the algorithm. Instead, we will make use of the following dataset introduced by McInnes:

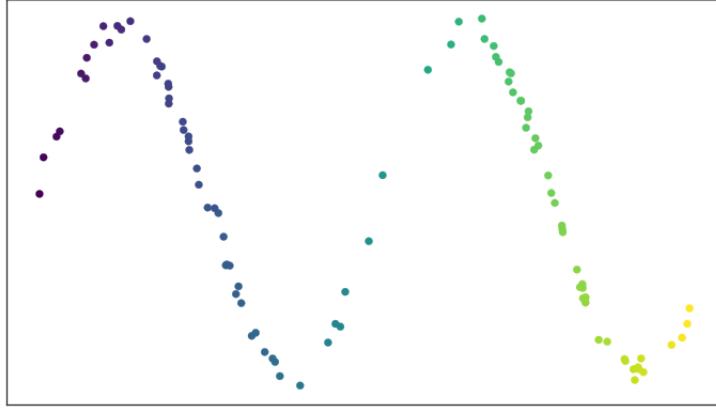


Figure 2.2: The dataset has been sampled with noise from a sine wave, McInnes (2018).

This given dataset (S, d) consists of a finite subset of \mathbb{R}^2 along with the restriction of the standard Euclidean metric. The underlying manifold \mathcal{M} , from which the dataset was sampled with noise, is a sine wave one-dimensional manifold.

Suppose our initial task is to construct a simplicial complex that accurately represents \mathcal{M} . To do this, we need to create an appropriate open cover of our dataset. A sufficiently refined cover ensures that the generated Čech complex $C_\epsilon(S)$ (Example 1.3.25), will exhibit the topological structure of the manifold. Corollary 1.3.30 is the justification of why this built simplicial complex accurately embodies the topology of the open cover.

We begin by placing open balls of a fixed radius at each data point to build a suitable open cover. However, we quickly realize that the choice of radius is critical. Suppose we set a ball radius equal to 1, obtaining the following picture:

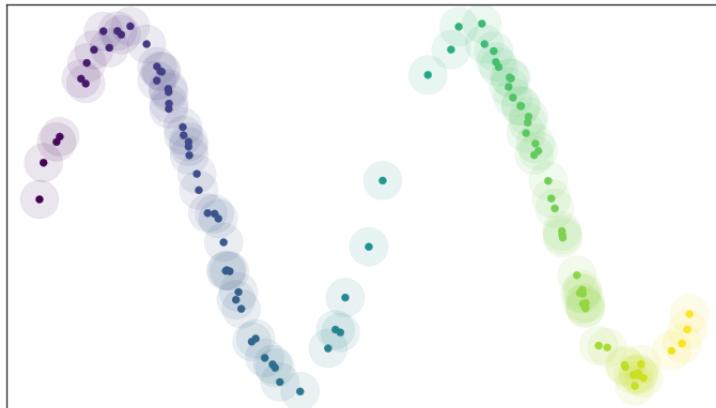


Figure 2.3: Open cover of the dataset, McInnes (2018).

and then create its respective Čech complex:

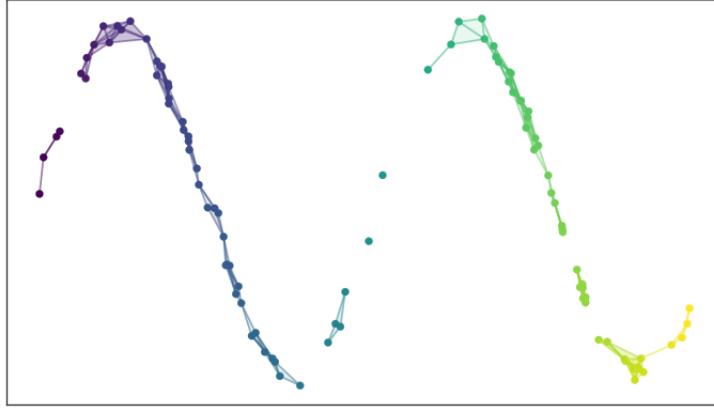


Figure 2.4: Čech complex of the dataset, McInnes (2018).

Notice that, even though we obtained an open cover of the dataset, we did not get an open cover of the underlying manifold. Consequently, we did not obtain a Čech complex that accurately represents \mathcal{M} . This strategy was unsuccessful because, as seen on the picture, there were regions with gaps and others where data points were too crowded together. Selecting a larger fixed radius would not have provided a correct solution either. This is because, although it would have created an open cover of \mathcal{M} , the resulting simplicial complex would not have accurately resembled the actual manifold, consisting instead of just a few high-dimensional simplices. In order to solve this problem, we require the following assumption:

Assumption 2: data is approximately¹uniformly distributed on the manifold

This assumption is not explained in the UMAP paper and can be misinterpreted. This statement should not be confused with assuming that the dataset was sampled i.i.d. uniformly from the manifold.

Assume our manifold \mathcal{M} is Riemannian (Definition 1.2.3), meaning it has a Riemannian metric g on \mathcal{M} . This allows us to compute the geodesic distance $d_{\mathcal{M}}$ between any two points in \mathcal{M} . Then, one possible mathematical way to express this assumption would be: $\forall y \in \mathcal{M}$ there exists $x \in S$ and real numbers δ_1, δ_2 satisfying $\delta_2 \leq \delta_1$, such that $d_{\mathcal{M}}(y, x) < \delta_1$ and $d_{\mathcal{M}}(x_i, x_j) > 2\delta_2$ for all distinct $x_i, x_j \in S$.

It is important to note that there are two metrics: the induced metric from the inclusion into the Euclidean space, as well as an unknown Riemannian metric with respect to which we have this uniform distribution of samples.

Notice that by assuming that the data is approximately uniformly distributed, we can now approximate the geodesic distance $d_{\mathcal{M}}$, so that it satisfies this assumption. When using

¹In the UMAP paper, Assumption 2 is stated as “data is uniformly distributed on the manifold”. However, uniform distribution cannot be exactly achieved for dimensions higher than one.

the Euclidean metric from the ambient space, we clearly observe that the data points are not uniformly distributed on the manifold. Nonetheless, by approximating a specific $d_{\mathcal{M}}$ we can achieve, for example with a fixed radius of 1, the following representation:

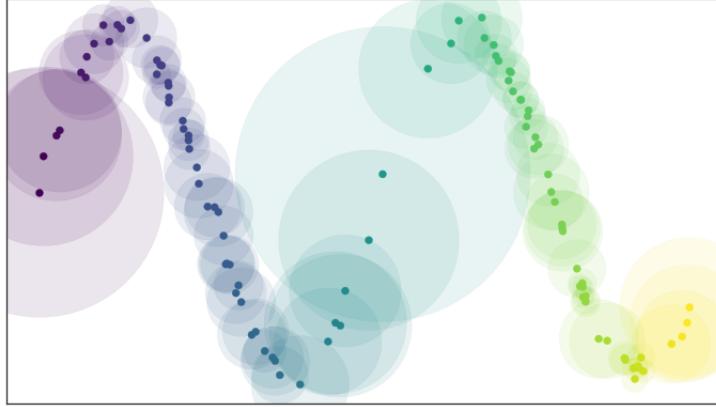


Figure 2.5: Open cover of the dataset considering a specific geodesic distance, McInnes (2018).

It is important to remark that each of these balls has a radius of approximately 1, with respect to the geodesic distance $d_{\mathcal{M}}$ we want to approximate. This picture indicates that, with an appropriate approximation of $d_{\mathcal{M}}$, we can find a suitable ball radius to construct an effective open cover, whose Čech complex accurately represents \mathcal{M} .

It is time now to explain how to approximate $d_{\mathcal{M}}$.

2.2.2 Riemannian manifold approximation

In order to approximate distances in our Riemannian manifold \mathcal{M} we will make use of the following lemma:

Lemma 2.2.1. *Let (\mathcal{M}, g) be a Riemannian manifold in an ambient Euclidean space \mathbb{R}^n , and let $p \in \mathcal{M}$ be a point. If we assume g is locally constant about p in an open neighbourhood U such that g is a constant diagonal matrix in ambient coordinates, then in a ball $B \subseteq U$ centered at p with volume $\frac{\pi^{n/2}}{\Gamma(n/2+1)}$ with respect to g , the geodesic distance from p to any point $q \in B$ is approximately $\frac{1}{r}d_{\mathbb{R}^n}(p, q)$ where r is the radius of the ball in the ambient space and $d_{\mathbb{R}^n}$ is the Euclidean metric.*

Proof. Refer to McInnes et al. (2018), Appendix A. □

This lemma is implying the following: when considering the tangent space of \mathcal{M} about p , we have a well-defined inner product which can be represented as an $n \times n$ matrix A

that, in general, depends on p . However, saying that g is locally constant about p in U , means that this matrix is independent of p . In fact, it is a diagonal matrix with

$$A_{ij} = \begin{cases} t & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases}$$

where $t \in \mathbb{R}$. Thus, in a sufficiently small ball $B \subseteq U$, we can approximate distances by scaling the Euclidean space \mathbb{R}^n with a factor of the radius of the ball in the Euclidean space.

We will add to our assumptions that our Riemannian metric g satisfies the conditions of the lemma, so that the lemma can help us approximate the geodesic distance $d_{\mathcal{M}}$.

Given that the data is uniformly spread across \mathcal{M} with respect to g , it follows that, except near the boundaries, any ball of fixed radius should contain a similar number of data points. With a slight change in perspective, a ball centered at an arbitrary data point containing k -nearest neighbours for some natural number k should be of similar radius, regardless of the data point. The reason for this new point of view is that it is empirically easier to choose the number of neighbours than to choose the correct size of a ball to define an open cover whose Čech complex accurately represents the underlying manifold \mathcal{M} . Furthermore, notice that the value of k is also significant. Choosing a smaller k would help us construct a finer open cover, whereas choosing a larger k would assist us in capturing the entire underlying manifold.

By Lemma 2.2.1, for a small enough k which gives us a sufficiently small ball B , we can approximate $d_{\mathcal{M}}$ from any data point to any of its k -nearest neighbours. Nonetheless, it is in this part of the algorithm where the mathematical foundations are completely overridden by heuristics. In UMAP’s implementation, k is a hyperparameter which, if not chosen sufficiently small, may invalidate Lemma 2.2.1. Moreover, it also permits setting a different “metric”² instead of the Euclidean, further disregarding Lemma 2.2.1 which approximates the geodesic distance using a scaled version of the Euclidean metric.

Even though there is a clear disconnection between UMAP’s mathematical ideas and its actual algorithm, we will proceed with our explanation since we consider the ideas that inspired the algorithm to be interesting and worth discussing.

The approximation of $d_{\mathcal{M}}$ is done as follows: given input hyperparameters k and a “metric” d , for each data point $x_i \in S$ we compute the set $\{x_{i_1}, \dots, x_{i_k}\} \subseteq S$ of the k -nearest neighbours of x_i under the “metric” d . This computation can be carried out using any

²UMAP also allows, as a hyperparameter, similarity functions that are not metrics, such as the cosine similarity. These functions may output negative values.

nearest neighbour algorithm. Then, for each x_i , we define

$$\rho_i := \min\{d(x_i, x_{i_j}) : 1 \leq j \leq k, d(x_i, x_{i_j}) > 0\}$$

which is the distance to the nearest neighbour of x_i . We set σ_i to be the value such that

$$\log_2(k) = \sum_{j=1}^k \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right).$$

The purpose of this ρ_i , which is used to define σ_i , is a heuristic approach to filter out pathological topological spaces that we would not normally encounter in real datasets, but could disrupt the correct behaviour of UMAP. This was asserted by McIness (2018), during an early presentation of the algorithm.

The parameter σ_i will play the role of r , the radius of the ball in the ambient space, in Lemma 2.2.1. The reason for defining σ_i in this way appears to be to mitigate the effect of having the k -th nearest neighbour very distant whereas the $(k - 1)$ -th nearest neighbours are clustered near x_i . This was also noticed by Jackson (2019). If that is the case, notice that, given the negative exponent, the k -th nearest neighbour will not contribute much to the sum. Moreover, the max function is required since d may output negative values.

By following this procedure and assuming that d is in fact a metric, we obtain for each data point x_i an approximation of the geodesic distance local to the point x_i . With this knowledge, we can construct, for each i , a finite metric space $(S, d_{\mathcal{M}_i})$ where $d_{\mathcal{M}_i}(x_i, x_j) := \frac{1}{\sigma_i} d(x_i, x_j)$ is the approximated geodesic distance local to x_i . Notice that here is another inconsistency: the local metric at x_i is defined using the information of the k -nearest neighbours respect to the input metric d . However, when defining $(S, d_{\mathcal{M}_i})$, we are using the same information to define, for example, the distance $d_{\mathcal{M}_i}(x_i, x_{i_{k+1}})$ where $x_{i_{k+1}}$ is the $(k + 1)$ -nearest neighbour of x_i . That is to say, we are using information about k -nearest neighbours to define the distance to the q -nearest neighbours with $q > k$, which is definitely conflicting. Additionally, we are defining the approximated geodesic distance local to x_i between points x_j and x_k with i, j, k distinct. This is definitely sub-optimal since $d_{\mathcal{M}_i}(x_j, x_k)$ is not something we would be interested in, considering that $d_{\mathcal{M}_j}(x_j, x_k)$ and $d_{\mathcal{M}_k}(x_j, x_k)$ would probably be better approximations. Nonetheless, we continue with the explanation acknowledging these issues.

Notice now that, in this context, we cannot define $d_{\mathcal{M}}$ in a natural way, since for any data points x_i, x_j there would be two possible distinct values for $d_{\mathcal{M}}$: $d_{\mathcal{M}}(x_i, x_j) = \frac{1}{\sigma_i} d(x_i, x_j)$ or $d_{\mathcal{M}}(x_i, x_j) = \frac{1}{\sigma_j} d(x_i, x_j)$. In other words, our local metrics cannot be “glued” into a

compatible global geodesic distance $d_{\mathcal{M}}$. Thus, the most we can do is to create finite metric spaces each with its own local metric.

UMAP's approach to solving this compatibility issue is to construct finite extended-pseudo-metric spaces (Definition 1.1.2) from the finite metric spaces and map them into finite fuzzy simplicial sets. These finite fuzzy simplicial sets (Example 1.4.64) allow for a natural gluing, helping to create a compatible global structure. We proceed to the next subsection to explain this in detail.

2.2.3 Fuzzy topological representation

Since in this subsection, unless otherwise stated, we will not use the dataset sampled from a sine wave for our explanation, we will use X to refer to a generic dataset.

In order to solve our compatibility issues we are going to map our finite extended-pseudo-metric spaces, constructed from our finite metric spaces, into finite fuzzy simplicial sets. These objects have the property that they are easy to merge into a global structure, unlike our finite metric spaces. However, notice that it is not enough to define an arbitrary functor from the category of finite extended-pseudo-metric spaces $FINEPMET$ (Example 1.4.10) to the category of finite fuzzy simplicial sets $FIN-SFUZZ$ (Example 1.4.64). There should exist some notion of “equivalence” between these categories, so that finite fuzzy simplicial sets appropriately represent finite extended-pseudo-metric spaces. This “equivalence” will be embodied by the concept of adjunction (Definition 1.4.34). With this in mind, we define the first functor (Definition 1.4.15) of our adjunction:

Definition 2.2.2. Define³ the functor $FinReal : FIN-SFUZZ \rightarrow FINEPMET$ by setting

$$FinReal(\Delta_{\leq a}^n) := (\{x_0, x_1, \dots, x_n\}, d_a)$$

where

$$d_a(x_i, x_j) := \begin{cases} -\log_2(a) & \text{if } i \neq j \\ 0 & \text{otherwise,} \end{cases}$$

and then extending it to a general finite fuzzy simplicial set X by

$$FinReal(X) := \operatorname{colim}_{\Delta_{\leq a}^n \rightarrow X} FinReal(\Delta_{\leq a}^n).$$

The action of $FinReal$ on a map $\Delta_{\leq a}^n \rightarrow \Delta_{\leq b}^m$ where $a \leq b$ defined by $\sigma : [n] \rightarrow [m]$ in Δ ,

³In the UMAP paper, there are minor inconsistencies in this definition, which we have fixed.

is given by the non-expansive map

$$(\{x_0, x_1, \dots, x_n\}, d_a) \mapsto (\{x_{\sigma(0)}, x_{\sigma(1)}, \dots, x_{\sigma(n)}\}, d_b).$$

Notice that this map is non-expansive since $a \leq b$ implies $d_a \geq d_b$. Moreover, σ does not have to be injective, so when $m < n$, there will be at least one pair of indices $i \neq j$ such that $\sigma(i) = \sigma(j)$.

In order to understand how this functor was formulated, we should understand what inspired it. In Spivak (2009), David Spivak defined the functor $Real : SFUZZ \rightarrow EPMET$ which maps the representable sheaf $\Delta_{\leq a}^n$ (Example 1.4.64) to the extended-pseudo-metric space

$$\left\{ (t_0, t_1, \dots, t_n) \in \mathbb{R}^{n+1} : \sum_{i=0}^n t_i = -\log_2(a), t_i \geq 0 \right\}$$

with the restriction of the standard Euclidean metric. A morphism $\Delta_{\leq a}^n \rightarrow \Delta_{\leq b}^m$ exists only if $a \leq b$ and is determined by a morphism $\sigma : [n] \rightarrow [m]$ in Δ . The action of $Real$ on such a morphism is given by the non-expansive map

$$(x_0, x_1, \dots, x_n) \mapsto \frac{\log_2(b)}{\log_2(a)} \left(\sum_{i_0 \in \sigma^{-1}(0)} x_{i_0}, \sum_{i_1 \in \sigma^{-1}(1)} x_{i_1}, \dots, \sum_{i_m \in \sigma^{-1}(m)} x_{i_m} \right).$$

The action of $Real$ on a morphism $\Delta_{\leq a}^n \rightarrow \Delta_{\leq b}^m$ is non-expansive because $0 < a \leq b \leq 1$ implies $\frac{\log_2(b)}{\log_2(a)} \leq 1$.

Then, Spivak extended this to a general simplicial set X by

$$Real(X) := \text{colim}_{\Delta_{\leq a}^n \rightarrow X} Real(\Delta_{\leq a}^n).$$

There are a couple of points to explain about Spivak's definition. We proceed with these explanations:

- The reason to use $-\log_2(a)$ is due to the fact that the membership strength is bounded between 0 and 1. Although another function could have also been used (such as another logarithm or also $\frac{1-a}{a}$), this one fulfills its role: mapping $(0, 1]$ to $[0, \infty)$. This is useful since $[0, \infty)$ is the appropriate interval to express distances in \mathbb{R}^{n+1} .
- The degree of membership of a fuzzy simplex is inversely correlated to the distance between points in its corresponding extended-pseudo-metric space. In other words,

a fuzzy simplex with a high degree of membership (“close” to 1) is mapped to a extended-pseudo-metric space with the shape of a “small” simplex. Similarly, a fuzzy simplex with a low degree of membership is mapped to a extended-pseudo-metric space with the shape of a “big” simplex.

- σ is a non-strictly order preserving map. So, it does not have to be injective. The sums indexed by preimages contemplate this situation.
- The category $EPMET$ is closed under colimits (Definition 1.4.41). This can be seen in Spivak (2009), Lemma 2.2 .
- The extension to a general simplicial set X is achieved by using colimits, which we learnt are a tool for merging objects. In this case, we are combining the extended-pseudo-metric spaces $Real(\Delta_{\leq a}^n)$, for the sheaves $\Delta_{\leq a}^n$ that have a morphism pointing to X .

Since the goal of UMAP is to reduce the dimension of datasets, which are finite metric spaces, there was a need to adapt Spivak’s definitions to acknowledge this finiteness. Nonetheless, the essence of Spivak’s ideas remains intact, explaining the definition of our functor $FinReal$, which is a slight modification of $Real$.

In the UMAP paper it is stated that $FinReal$ preserve colimits, fulfilling a necessary condition to be a left adjoint. This assertion is a consequence of $Real$ preserving colimits, as shown in Spivak (2009). Hence, we define the functor $FinSing$ and show it is the right adjoint of $FinReal$. This functor is again an adaptation of a functor called $Sing$ from Spivak’s paper. The only difference between them is that McIness replaced $Real$ with $FinReal$ in the definition of $FinSing$.

Definition 2.2.3. Define the functor $FinSing : FINEPMET \rightarrow FIN-SFUZZ$ by

$$\begin{aligned} FinSing(Y) : (\Delta \times OPEN(I))^{op} &\rightarrow SET \\ ([n], [0, a)) &\mapsto hom_{FINEPMET}(FinReal(\Delta_{\leq a}^n), Y) \end{aligned}$$

for $Y \in \mathcal{O}_{FINEPMET}$.

This definition is telling us that, in order to construct a finite fuzzy simplicial set that corresponds to a finite extended-pseudo-metric space, each of its fuzzy simplices should model how the objects $FinReal(\Delta_{\leq a}^n)$ and Y are related within the category $FINEPMET$. Since the construction of a finite fuzzy simplicial set via $FinSing$ is determined by the functor $FinReal$, it is expected that $FinReal$ and $FinSing$ form an adjunction. Hence, we have the following theorem:

Theorem 2.2.4. *The functors $\text{FinReal} : \text{FIN-SFUZZ} \rightarrow \text{FINEPMET}$ and $\text{FinSing} : \text{FINEPMET} \rightarrow \text{FIN-SFUZZ}$ form an adjunction with FinReal the left adjoint and FinSing the right adjoint.*

Proof. Refer to McInnes et al. (2018), Appendix B. □

Intuitively, finite extended-pseudo-metric spaces and finite fuzzy simplicial sets are different ways to express distances between our data points. Moreover, FinReal and FinSing are the tools that allow us to change between these two perspectives.

Notice that these functors require finite extended-pseudo-metric spaces instead of the finite metric spaces we constructed at the end of Subsection 2.2.2. Thus, it is time now to construct these new objects.

For each data point of our dataset, we construct a finite extended-pseudo-metric space with its own local notion of distance as follows:

Let $X = \{x_1, \dots, x_n\}$ be a dataset in \mathbb{R}^n . For each $x_i \in X$, we find any extended-pseudo-metric spaces $\{X, d_i\}$ that satisfies

$$d_i(x_j, x_k) := \begin{cases} d_{\mathcal{M}_i}(x_j, x_k) - \rho_i & \text{if } i = j \text{ or } i = k \text{ but not } i = j = k \\ 0 & \text{if } i = j = k, \end{cases}$$

where ρ_i is the distance to the nearest neighbour of x_i and $d_{\mathcal{M}_i}$ is the approximated geodesic distance local to x_i , as defined in Subsection 2.2.2.

In the UMAP paper, the definition above presents some inconsistencies, which is why we have reformulated it completely. We proceed to explain the irregularities found in the paper. First, the case where $i = j = k$ appears to have been inadvertently overlooked. Then, it defines $d_i(x_j, x_k) := \infty$ for distinct i, j, k . What this problematic definition wants to imply is that $d_i(x_j, x_k)$ for distinct i, j, k is “irrelevant”, given that $d_j(x_j, x_k)$ and $d_k(x_j, x_k)$ are better approximations since they are defined using the local notions of distances with respect to x_j and x_k , respectively. The issue with UMAP’s definition, is that it does not satisfy the triangle inequality since, for distinct i, j, k , we have

$$\begin{aligned} \infty &= d_i(x_j, x_k) \\ &\leq d_i(x_j, x_i) + d_i(x_i, x_k) \\ &< \infty, \end{aligned}$$

which is a contradiction.

Recall that our problem of last subsection was that these local notions of distance may not be globally compatible, meaning that $d_{\mathcal{M}_i}(x_i, x_j)$ may be different from $d_{\mathcal{M}_j}(x_i, x_j)$. We are now able to solve this issue by mapping the newly constructed finite extended-pseudo-metric spaces into finite fuzzy simplicial sets, which can be “glued” into a compatible global structure via some choice of fuzzy union.

Although UMAP’s theoretical ideas consider the possibility that our finite fuzzy simplicial sets may have higher dimensional simplices, its current implementation only allows the existence of 0-simplices (vertices) and 1-simplices (edges). This means that after mapping any finite extended-pseudo-metric space via *FinSing*, the n -simplices with $n \geq 2$ are discarded. The reason for this choice is to make the algorithm less computationally expensive. Additionally, for each $\text{FinSing}((X, d_i))$, we also dismiss the edge between x_j and x_k for all distinct i, j, k . This is because we are only interested in the edge between x_j and x_k of each of the finite fuzzy simplicial sets $\text{FinSing}((X, d_j))$ and $\text{FinSing}((X, d_k))$ defined functorially with respect to the local metrics at x_j and x_k , respectively. One way to express the discarding of a fuzzy simplex σ in $\text{FinSing}((X, d_i))$ is by modifying the respective membership function μ_i to 0 when applied to the simplex. Formally, this implies defining a new membership function μ_i^* on $\text{FinSing}((X, d_i))$ which is 0 at the simplices we want to discard and equal to μ_i otherwise. In what follows, for each i , we will redefine $\text{FinSing}((X, d_i))$ to denote the modified finite fuzzy simplicial set that contemplates the discarding of the respective simplices.

For each finite extended-pseudo-metric space, we then obtain a possibly distinct graph. If we put all these graphs on top of each other, we see that there might be two edges, with different degrees of membership, between any two points. This can be visualized with the following picture of the example from previous subsections:

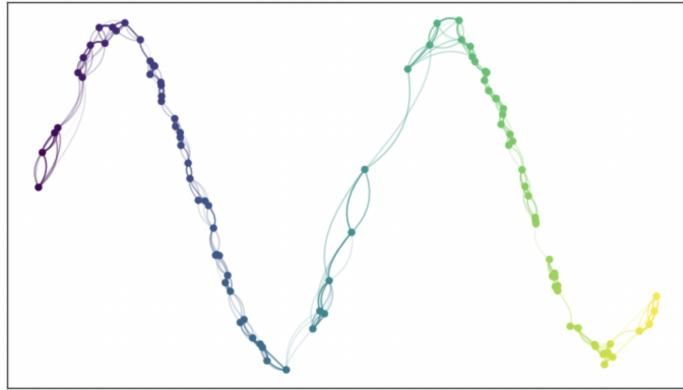


Figure 2.6: Overlapping graphs of finite fuzzy simplicial sets, McInnes (2018).

We are going to construct a new finite fuzzy simplicial set denoted as $\text{FinSing}(X)$ which

combines the finite fuzzy simplicial sets $\text{FinSing}((X, d_i))$ for each i .

Given that these finite fuzzy simplicial sets only contain vertices and edges, and at most two edges between any two distinct data points have a degree of membership greater than zero, there is a natural choice of fuzzy union in UMAP. If we think of the degree of membership of an edge as the probability that the edge exists, the combined degrees of membership of two edges between two data points can be seen as representing the probability that at least one of the two edges exist. So, let α be the degree of membership of the edge between data points x_i and x_j from the graph given by $\text{FinSing}((X, d_i))$, and let β be the degree of membership of the edge between data points x_i and x_j from the graph given by $\text{FinSing}((X, d_j))$. We define the degree of membership of the unique edge between data points x_i and x_j in $\text{FinSing}(X)$ to be $\alpha + \beta - \alpha \cdot \beta$.

By applying this fuzzy union to each edge, we obtain a graph which has at most one edge between two vertices. This can be visualized with the following picture:

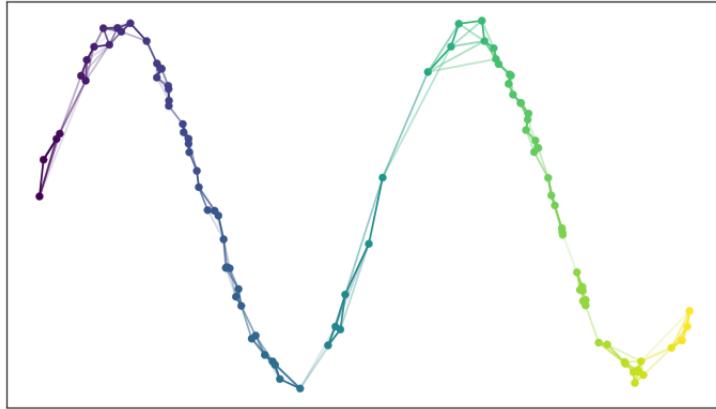


Figure 2.7: Fuzzy union of finite fuzzy simplicial sets, McInnes (2018).

It is time now to proceed to the final subsection of UMAP.

2.2.4 Optimizing a lower-dimensional representation

We have reached a point where we understand how to convert our high-dimensional dataset $X = \{x_1, x_2, \dots, x_n\} \subseteq \mathbb{R}^n$ into a finite fuzzy simplicial set $\text{FinSing}(X)$. We are now interested in obtaining a lower-dimensional representation of X .

Let $Y = \{y_1, y_2, \dots, y_n\} \subseteq \mathbb{R}^m$ with $m < n$ be some dataset such that each y_i represents one of the x_i . That is, there exists a bijective function $f : X \rightarrow Y$ such that $f(x_i) = y_i$. We will call Y a low-dimensional representation of X . The dataset Y may be an arbitrary chosen lower-dimensional dataset or may be randomly initialized. By following the same

procedure that we used to convert X into the finite fuzzy simplicial set $\text{FinSing}(X)$, we can transform Y into the finite fuzzy simplicial set $\text{FinSing}(Y)$. Unlike X where we were interested in finding the Riemannian manifold approximation from which it was sampled, the target manifold of Y is chosen a priori and will usually be the lower-dimensional Euclidean space \mathbb{R}^m .

We now have two finite fuzzy simplicial sets $\text{FinSing}(X)$ and $\text{FinSing}(Y)$ which consist only of 0-simplices and 1-simplices. To measure how similar the topological structures of $\text{FinSing}(X)$ and $\text{FinSing}(Y)$ are, UMAP uses fuzzy set cross-entropy. Before defining it, first notice that each edge in $\text{FinSing}(X)$ has some degree of membership. Hence, we can define (E_X, μ) to be the fuzzy set of all edges E_X of $\text{FinSing}(X)$ along with its membership function $\mu : E_X \rightarrow [0, 1]$ given by $\text{FinSing}(X)$. Similarly, we define (E_Y, v) for the fuzzy set corresponding to the edges of $\text{FinSing}(Y)$, where $v : E_Y \rightarrow [0, 1]$ is the membership function given by $\text{FinSing}(Y)$.

Since the cardinality of X and Y is the same, we can think of (E_X, μ) and (E_Y, v) as complete graphs whose probability that an edge exists is given by μ and v , respectively. A **complete graph** is an undirected graph in which every pair of distinct vertices is connected by a unique edge. Thus, since E_X and E_Y are the sets of edges of a complete graph with the same number of vertices, we can define $E := E_X = E_Y$, and also:

Definition 2.2.5. The **fuzzy set cross-entropy** C is defined as

$$C : \{(E, \mu, v) \mid \mu, v \text{ are membership functions defined on a set } E\} \rightarrow \mathbb{R}$$

such that

$$C(E, \mu, v) = \sum_{e \in E} \left(\mu(e) \log \left(\frac{\mu(e)}{v(e)} \right) + (1 - \mu(e)) \log \left(\frac{1 - \mu(e)}{1 - v(e)} \right) \right).$$

Our objective is now to minimize C . Let us unpack this cross-entropy definition to understand what it means to be minimized:

The first term $\mu(e) \log \left(\frac{\mu(e)}{v(e)} \right)$ of the sum tells us that if the probability $\mu(e)$ of the edge e existing in $\text{FinSing}(X)$ is high, we want the quotient $\frac{\mu(e)}{v(e)}$ to be as close to 1 as possible to minimize C . This would give us a first term close to 0. In order for the quotient to be close to 1, we need that the probability $v(e)$ of the edge e existing in $\text{FinSing}(Y)$ is also high. Intuitively, this term acts as an attractive force, pulling points together in the lower-dimensional representation if the probability of the edge existing in the higher-dimensional representation is high. On the other hand, the second term acts as a repulsive force, pushing points away in the lower-dimensional representation if the probability of

the edge existing in the higher-dimensional representation is low.

Notice that, by modifying Y , we change v , which in turn modifies C . Thus, the minimization of C can be achieved through this procedure.

Having just finished our explanation of the mathematical foundations of this state-of-the-art algorithm, we recognize that many aspects are unsatisfying due to numerous inconsistencies. Nevertheless, we still consider the thought process behind the algorithm to be appealing.

Now that we have examined PCA and UMAP, we will proceed to the next section where we will explore some applications of dimensionality reduction.

2.3 PCA and UMAP applications

The MNIST and the Fashion-MNIST datasets, introduced by LeCun et al. (2010) and Xiao et al. (2017) respectively, are some of the most iconic datasets. They are often used as benchmarks for testing different machine learning algorithms.

The MNIST dataset and the Fashion-MNIST dataset each consist of 60000 training images and 10000 testing images of handwritten digits and fashion images, respectively. We will now use these datasets to visualize how PCA and UMAP work. After that, we will proceed to the next section where we will explore another dimensionality reduction algorithm.

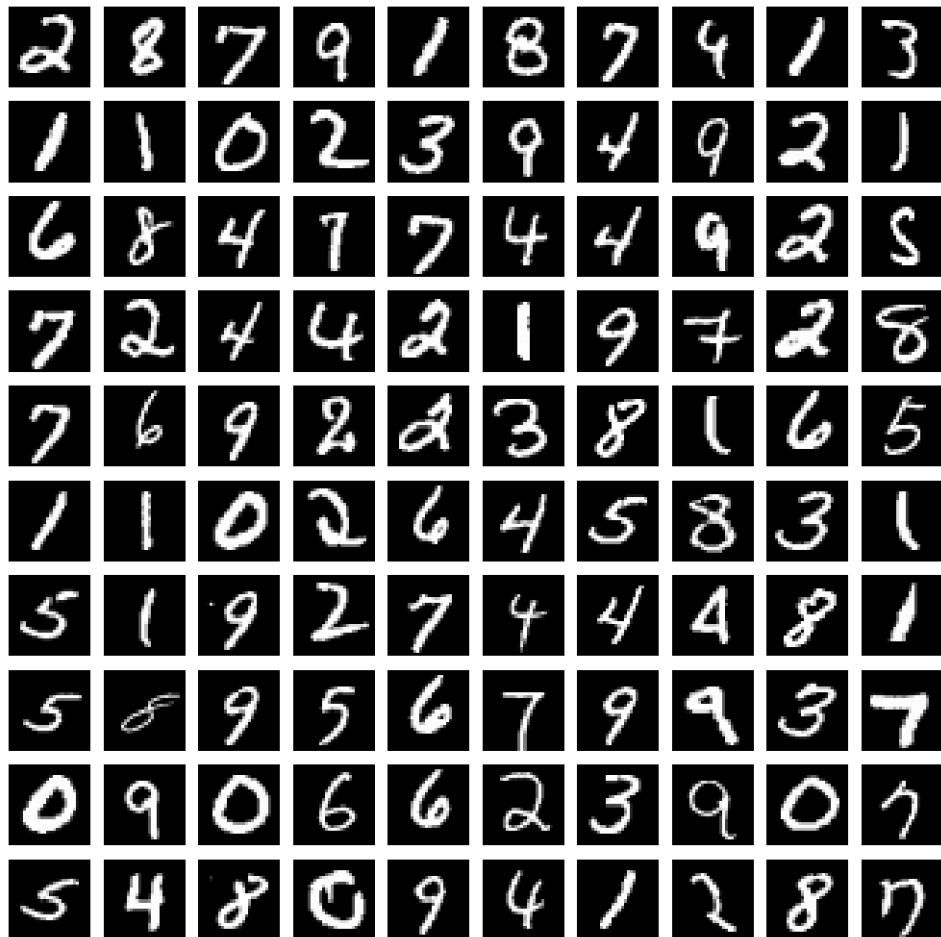


Figure 2.8: 100 images of the MNIST dataset, LeCun et al. (2010).

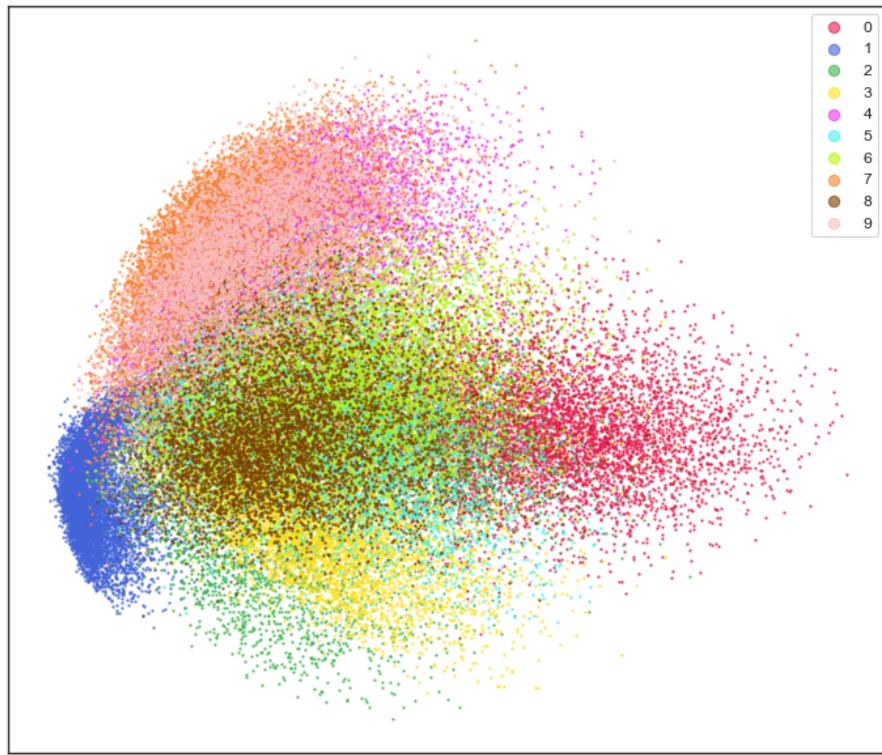


Figure 2.9: 2-dimensional PCA applied to the MNIST dataset.



Figure 2.10: 2-dimensional UMAP applied to the MNIST dataset.



Figure 2.11: 100 images of the Fashion-MNIST dataset, Xiao et al. (2017).

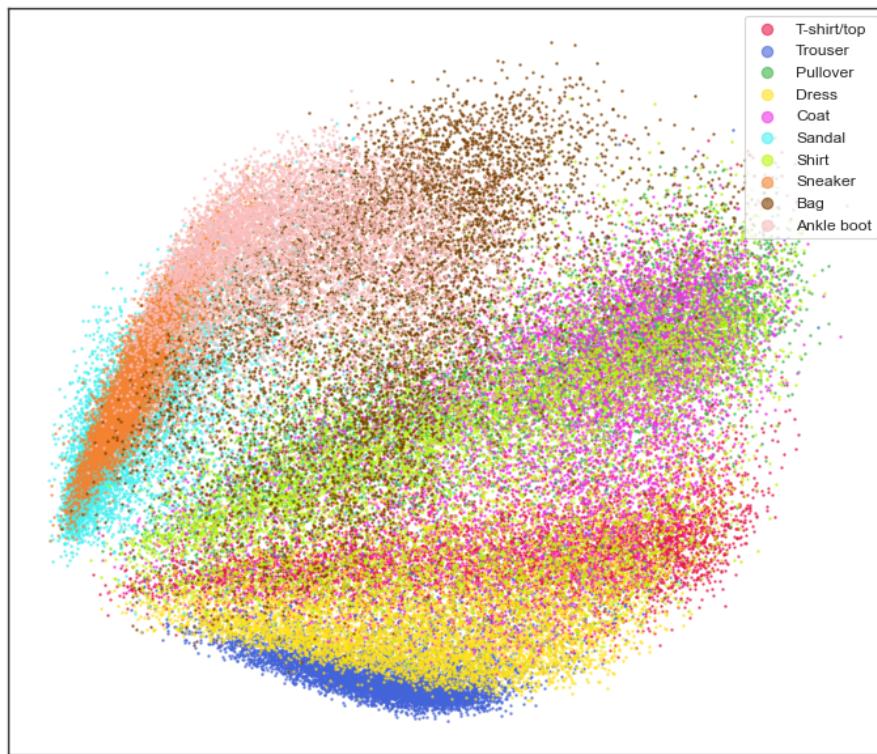


Figure 2.12: 2-dimensional PCA applied to the Fashion-MNIST dataset.

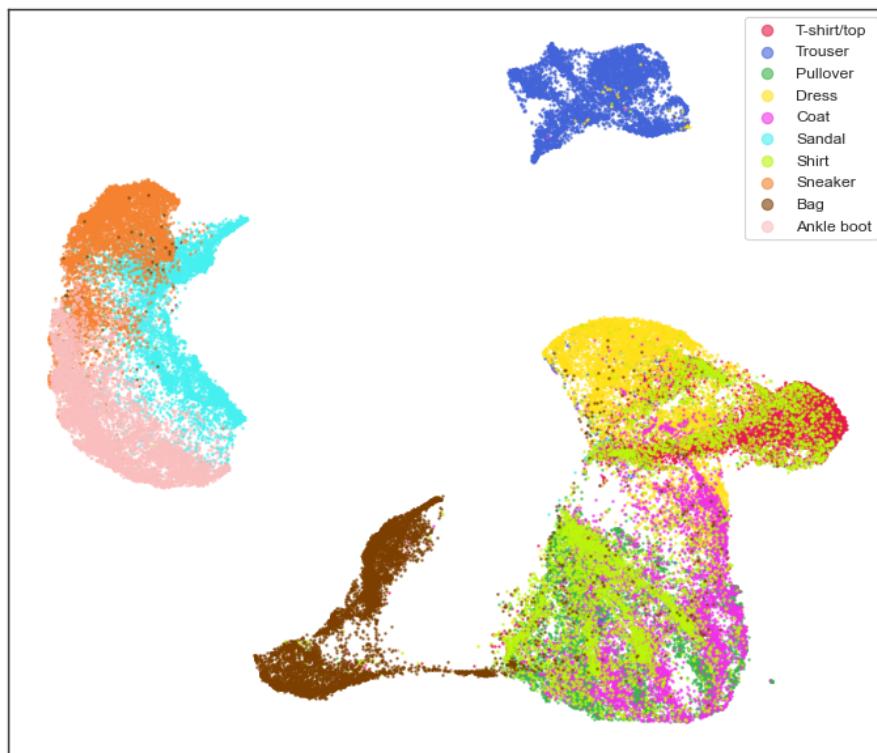


Figure 2.13: 2-dimensional UMAP applied to the Fashion-MNIST dataset.

2.4 Mapper

Mapper, introduced by Singh et al. (2007), is a simple yet powerful dimensionality reduction algorithm. In short, it takes a dataset as input, executes some straightforward computations, and outputs a simplicial complex that “accurately” represents the dataset. Since simplicial complexes are simple objects, their visualization is easily comprehensible, even for non-specialized audiences, making Mapper an invaluable tool. In our football application, we will use Mapper to leverage the strengths of PCA and UMAP.

PCA and UMAP are not restricted to data visualization. They are often used, for example, for data compression to improve the performance of machine learning models. Mapper, on the other hand, is mainly a visualization tool. This means that its goal is to highlight areas of interest, so that we can later dig deeper into them. We proceed to explain in detail how Mapper works:

In the first step, we require a dataset as input. In order to make this exposition as clear as possible, we will continually refer to a toy example that starts with the following dataset:

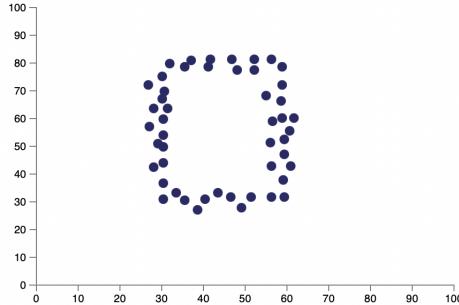


Figure 2.14: Dataset of a square with noise.

Notice that the human eye can easily recognize that this dataset was sampled from a square, even though it contains some noise.

In the second step, we need to define a **filter function** which projects the data points into a lower-dimensional space. This function can take many forms. It is in this step where our discussed dimensionality reduction algorithms merge since, in particular, we are allowed to use PCA or UMAP as the filter function to construct a lower-dimensional representation. For our toy example, we will use $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ as $f(x, y) = y$ as our filter function.

After projecting the data points, we define, for an index set A , an open cover $\{U_\alpha : \alpha \in A\}$ on this projection. In our example, we could have the following picture:

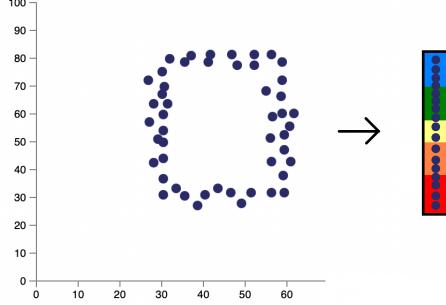


Figure 2.15: Projecting the dataset into the y coordinate and defining an open cover represented with colors.

In our example, the open cover consists of the open sets U_{blue} , U_{yellow} , and U_{red} , represented by their respective colors. The green and orange colors represent the overlaps between the respective open sets of the cover.

Then, we look at each of the sets $f^{-1}(U_\alpha)$ separately and apply a clustering algorithm of our choice to find related subsets on each preimage. In our example, we could obtain the following picture:

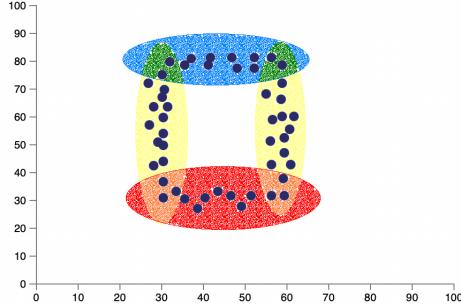


Figure 2.16: Clusters of the preimages after applying some clustering algorithm.

It is important to note that the clustering algorithm separated the yellow elements into two distinct sets.

Finally, we construct a simplicial complex by taking the nerve of the open cover (Definition 1.3.23) of our dataset. In our example, this would be:

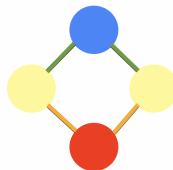


Figure 2.17: Nerve of the open cover.

Although Mapper's procedure is quite simple, it leaves many questions unanswered. This is because the queries "What filter function and clustering method should we use?" and "How should the open cover be defined?" do not have definitive answers. The output of the algorithm depends on hyperparameters that need to be set. Basically, we can define the filter function, clustering method, and open cover however we want. When exploring our football application, we will first start with some hypotheses about our dataset and then find sets of hyperparameters that satisfy them.

We will see that the combination of UMAP as the filter function along with agglomerative clustering provides one of the best representations of our football dataset that we could find. **Agglomerative clustering** is a simple clustering algorithm that starts by treating each data point as its own cluster. It then successively merges each cluster with the closest one until a predefined distance threshold is reached.

With all these dimensionality reduction algorithms at hand, it is time now to learn some football.

Football applications

In this chapter, we will explore football from a mathematical perspective. Given the nature of this thesis, we will discuss the most played sport in the world using a high-level analysis, rather than spending time on the code implementation and parameter details. We will see that by making use of some state-of-the-art tools, we will obtain plenty of non-trivial insights.

As stated in the introduction, our main goal is to understand our dataset in order to identify the important questions to ask. If the objective were solely to make predictions, employing several machine learning tools would definitely improve our forecasting. Nevertheless, our dataset contains a policy that forbids training any kind of machine learning model. This is why we have completely disregarded the use of neural networks, for example.

This chapter will examine three different applications:

- Understanding which features make a player more or less valuable.
- Generating a graph that clusters players according to their skills and performance.
- Predicting a team's performance after adding new players.

Before delving into our applications, we will first state the tools we will use.

3.0.1 Methodology

In the following analysis, we have used Python 3 (Van Rossum & Drake (2009)) in combination with the libraries Pandas (McKinney et al. (2010)), Numpy (Harris et al. (2020)), Matplotlib (Hunter (2007)), Sklearn (Pedregosa et al. (2011)), Seaborn (Waskom et al. (2017)), Plotly (Inc. (2015)), Igraph (Csardi & Nepusz (2006)), Gudhi (The GUDHI Project (2015)), UMAP (McInnes et al. (2018)), and Kmapper (van Veen et al. (2019)).

The dataset we have used belongs to FBREF (2023). It consists of football players from the 2022 – 2023 Big 5 European Leagues: Bundesliga, LaLiga, Ligue 1, Premier League, and Serie A.

We have selected each player that has played at least 1350 minutes during the season, along with all the 111 attributes available in our dataset for each player. Each feature has been normalized using its z-score.

The only assumption that we have made is that a player’s salary is completely determined by their football skills. Although this is a reasonable assumption to make, it should be noted that, in some cases, other external factors also contribute.

We are now ready to discuss our applications.

3.0.2 First application and dataset cleaning

The first step in any data science project is usually cleaning the dataset. To achieve this, we start by computing its correlation matrix, which can be visualized as follows:

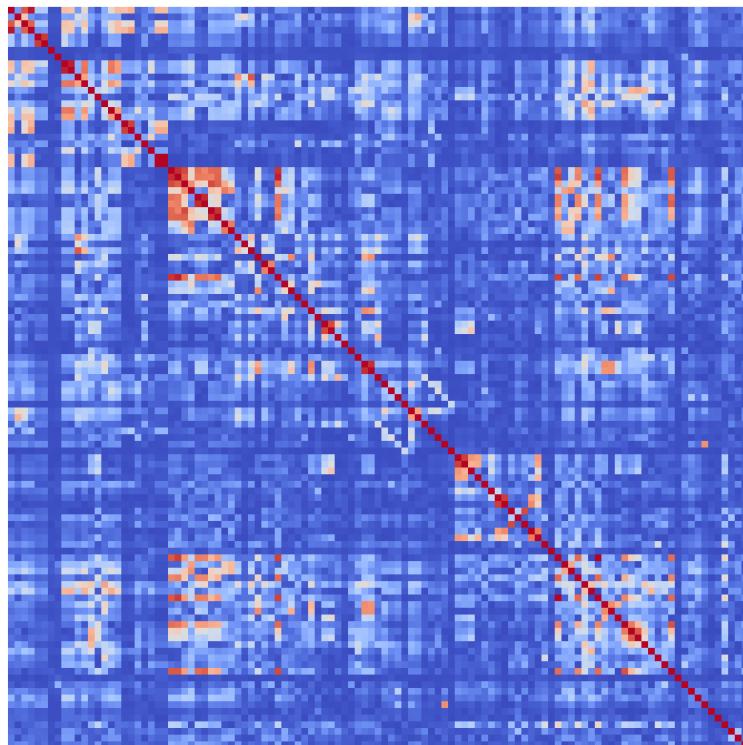


Figure 3.1: Correlation matrix of our football dataset.

This is a 111×111 symmetric matrix, where the (i, j) -entry represents the correlation between features i and j . The bluer the matrix entry, the less correlated the two variables are, whereas the redder, the more correlated the two variables are.

As a consequence of the high number of features, tagging each column/row with its feature name is not feasible, as the tags would not be easy to read on this thesis page. To better understand this picture, let us consider the following example: our dataset contains two features called “Passes completed” and “Short passes completed”. These two attributes are clearly highly correlated since short passes are also passes. In our correlation matrix, the intersection of these two variables would appear as a strong red square.

Before deleting some of the highly correlated variables, we can use the correlation matrix to determine which variables most positively or negatively contribute to a player’s salary. In order to do this, we look at the column representing the “Salary” and identify the highest and the lowest correlation values.

Remark 3.0.1. *A non-visual method for identifying features that significantly influence a player’s salary can be achieved by using the **Random Forest Algorithm**. This alternative method achieves higher precision by testing how many different combinations of features affect the target variable. On top of that, it is capable of capturing non-linear relationships between variables. However, as a consequence of our dataset restrictions, we only suggest that using this method is likely to provide “similar” results.*

In decreasing order of importance, some of the variables that make a player receive a better salary are:

- Sum of goals and assists.
- Completed passes between back defenders into open space.
- Passes received.

Similarly, some of the features that make a player receive a worse salary are:

- Number of times a player attempts but fails to win possession of the ball during an aerial duel with an opponent.
- Number of fouls committed.
- Number of times blocking the ball by standing in its path.

Although the importance of these attributes is well-known in football, understanding their exact ordering is a valuable yet simple insight.

It is evident that the correlation matrix has a high degree of entropy. Deleting correlated variables by just looking at this matrix can be a challenging task, given its chaotic appearance. To simplify this procedure, we will use the tools from Section 1.5.

We apply the Matrix Sorting Algorithm to enhance the visualization of our correlation matrix, obtaining the following sorted matrix:

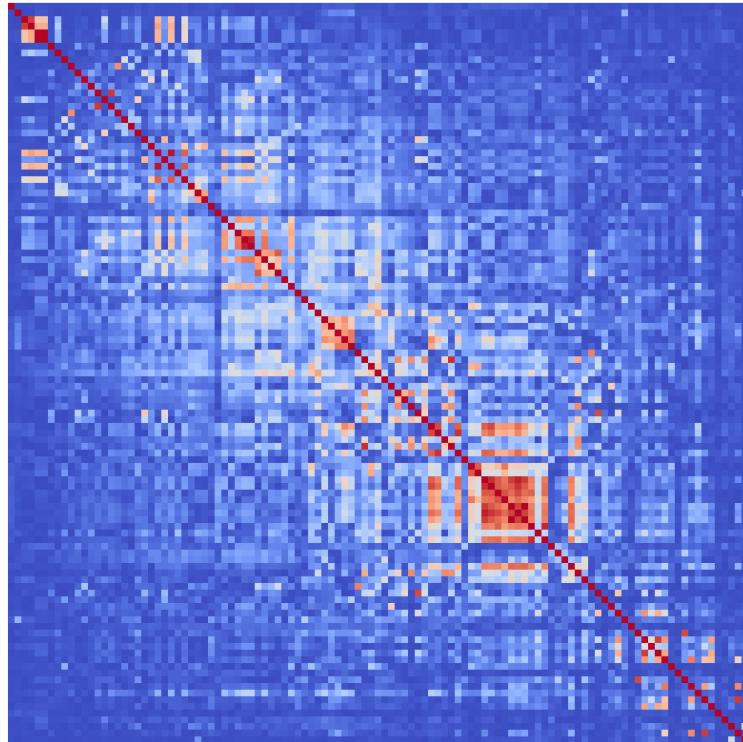


Figure 3.2: Correlation matrix of our football dataset after applying the Matrix Sorting Algorithm.

Notice that this modified correlation matrix highlights the points of interest, now represented as reddish squares.

We proceed to delete redundant highly correlated features. When removing correlated variables, it is important to also consider negative correlation, which can be addressed by considering the absolute values of the correlation matrix. Furthermore, we choose to drop the “Salary” feature since we assume that it is entirely a consequence of other variables. After taking these factors into account and choosing a threshold dependent on the dataset, the number of variables was reduced from 111 to 55. We now compute the correlation matrix of the pruned dataset:

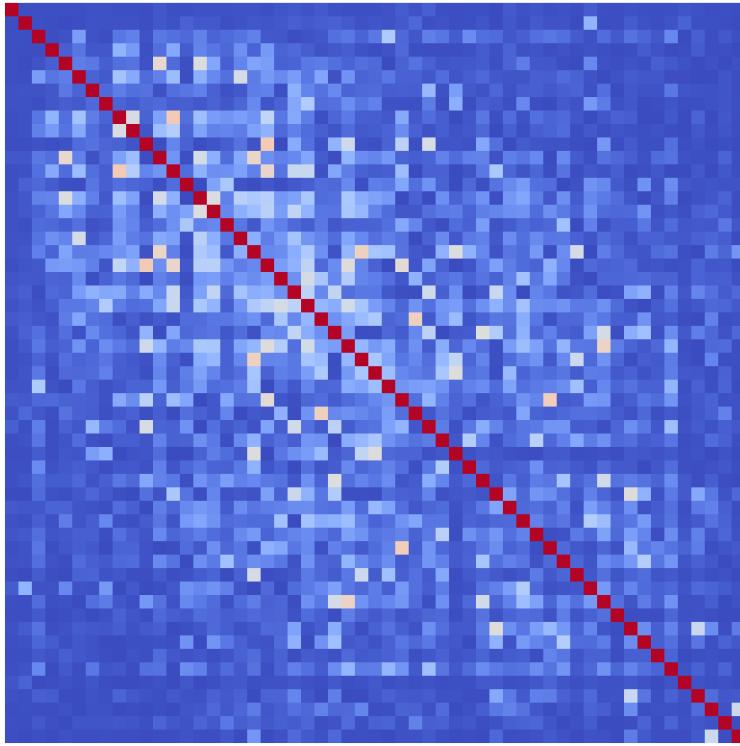


Figure 3.3: Sorted correlation matrix of our football dataset after deleting highly correlated variables.

Notice that all eye-catching red clusters from the last picture have disappeared, indicating that the dataset is now less correlated.

Having cleaned our dataset, we can now examine our most valuable applications.

3.0.3 Visualizing our dataset

As an initial approach, we proceed to visualize our dataset using PCA and UMAP, where the colors represent the different players' positions. These tags are:

- DF: defender.
- DF, MF: defender-midfielder.
- FW: forward.
- GK: goalkeeper.
- MF: midfield.
- MF, FW: midfielder-forward.

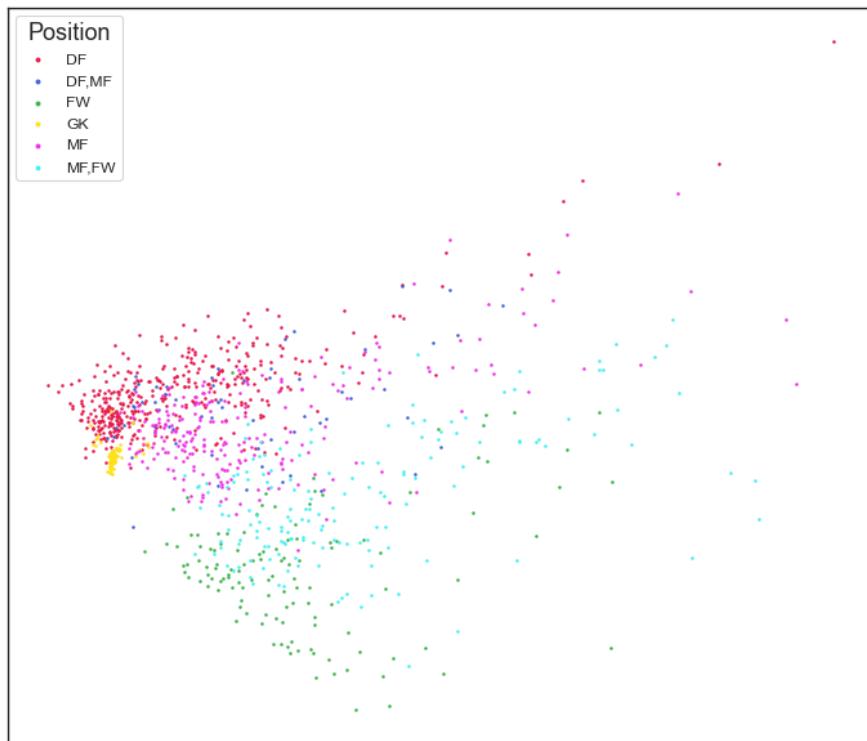


Figure 3.4: 2-dimensional PCA applied to our football dataset.

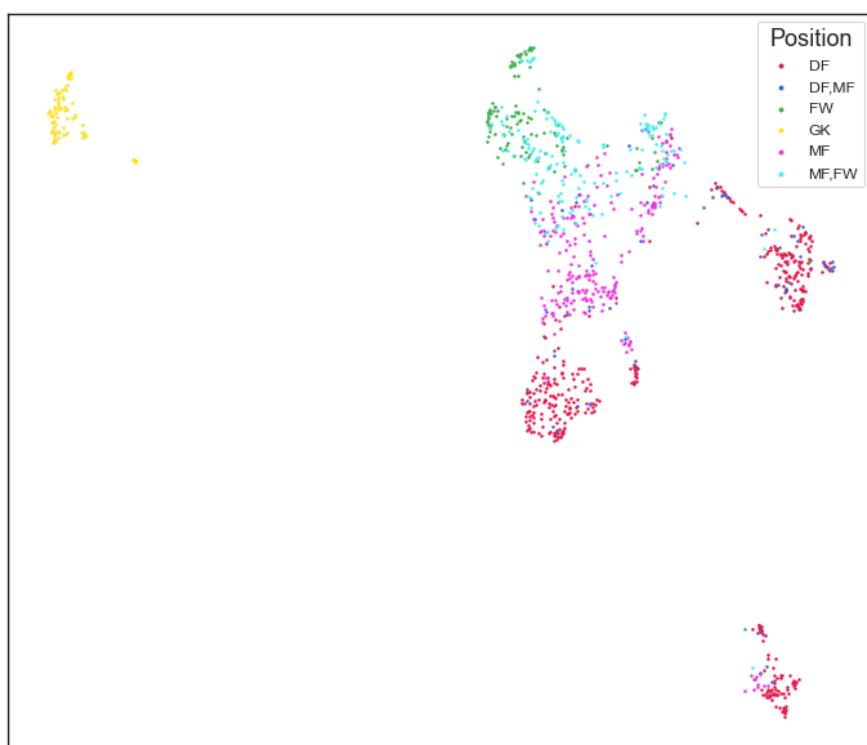


Figure 3.5: 2-dimensional UMAP applied to our football dataset.

Notice that, although UMAP does a better job than PCA in discriminating players' positions, the conclusions we can draw just by looking at these pictures are not very enlightening. We find it extremely difficult to obtain knowledge from the PCA picture. By looking at the UMAP picture, we definitely get more information: goalkeepers are far from the rest of the players, which is appropriate given that their playing style is completely different from that of field players. Moreover, there is a group of players of different positions in the bottom right, indicating some particularity. After inspecting this group, it remains arduous to make definitive assertions, other than the fact that they are more defensive than the average player. This is why we need to use Mapper.

In order to obtain a Mapper graph that accurately represents our dataset, we tested thousands of potential parameters and discarded those that did not satisfy our basic principles. These principles, which require elementary football knowledge, are the following:

- Goalkeepers should be in their own cluster, given that their playing style is radically different from that of field players.
- Most of the best players such as Lionel Messi and Neymar should be in their own cluster or at most sharing it with a few other players. Also, they should not have edges connecting them to regular players.

The Mapper combination that we have found to provide one of the best representations of our football dataset consists of UMAP as the filter function and agglomerative clustering as the clustering method. Agglomerative clustering allows us to predefine distance thresholds. This means that, if we set a very small distance parameter, each player will lie in their own cluster without any edges between them. On the other hand, if the distance parameter is too high, there will not be isolated clusters, failing to satisfy our second principle.

We now proceed to visualize our crafted Mapper graph, where the colors represent the average number of goals plus assists for each cluster. Hence, the bluer the cluster, the more defensive it is, while the redder the cluster, the more aggressive.

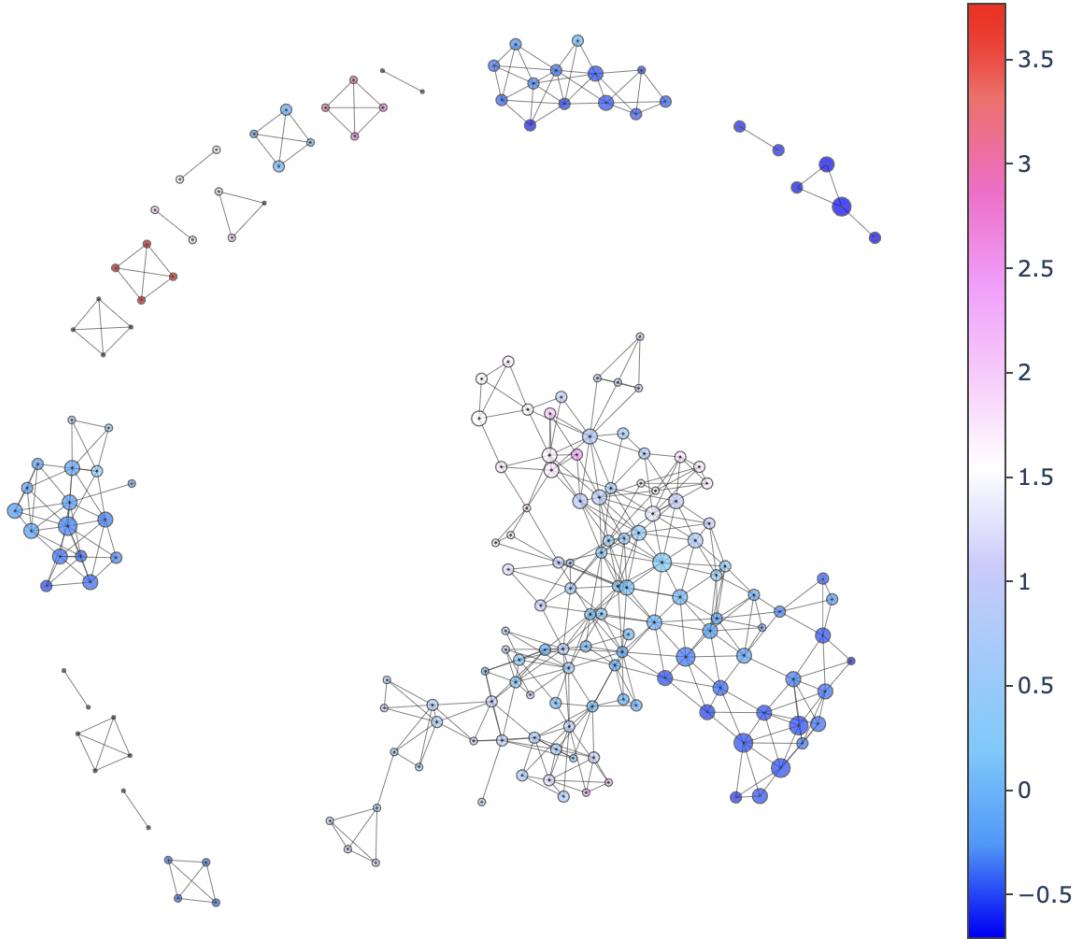


Figure 3.6: Football dataset after applying Mapper in combination with UMAP and agglomerative clustering.

By having players organized in clusters and connected with edges, it becomes a simple task to obtain insights. Our discoveries include:

- Lionel Messi, Neymar, Kevin De Bruyne, Toni Kroos, Karim Benzema and Erling Haaland are globally recognized as some of the finest players in the world. Our model indeed considers them as outstanding players. They share their cluster with none or few other players, validating the correctness of our model.
- Football players are bounded below. This assertion is expected, given that if a player performs worse than a threshold, he cannot play in the best leagues. On the other hand, a top player can be indefinitely good. This is reflected by the fact that most of the disconnected clusters contain the greatest players, whereas the ordinary ones lie in one of the multi-connected clusters on the bottom right of the graph.

- Domenico Berardi shares a cluster with Karim Benzema and Erling Haaland. The fact that Berardi's salary is less than 30% of each of the other two players is definitely striking. Moreover, we have found football articles such as Sharland (2022) recognizing Berardi as a highly underrated player.
- Teji Savanier shares a cluster with Paulo Dybala. Savanier's salary is less than 20% of Dybala's, yet they are similar players according to our model.
- Kieran Trippier lies alone in his cluster. This veteran seems to be a superlative player yet he flies under the radar of the top teams.

It is important to note that since the features were gathered individually for each different league, each conclusion is locally true. This means, for example, that Teji Savanier in Ligue 1 is similar to Paulo Dybala in Serie A.

Having presented some of the findings from our Mapper graph, we now move on to discuss our final application.

3.0.4 Predicting performance

Given that football is a team sport, analyzing players as individuals is just one factor that contributes to the performance of a team. There will be cases where adding a top player to a current team will not improve its functioning, especially if the team already has another player with similar skills. On the other hand, buying a regular player who satisfies the team's needs can be an outstanding decision.

Since football is an extremely complex game where each team requires many different skills to succeed, our initial hypothesis is that the more skill-diverse a team is, the better. To test this hypothesis, we select the top 3 and the bottom 3 teams of each of the five leagues and compute their *ST* (Definition 1.3.47), which will indicate the sparseness of a team. In order to do this, we first select the 11 players who have played the most minutes from each team. This is a necessary step because some teams might have more than 11 players exceeding our minutes threshold. We chose 11 because that is the number of starting players on the field per team in football. This approach ensures we are comparing the homology of teams with an equal number of players, which is critical since having different numbers of players may disrupt the analysis. After selecting the players, we compute the Vietoris-Rips filtration (Definition 1.3.12) for each team separately and determine its simplicial persistent homology. Then, we calculate the *ST* of each filtration. Finally, for illustration purposes, we compare the box plots of the *ST* of the best 15 teams and of the bottom 15 teams, obtaining the following graph:

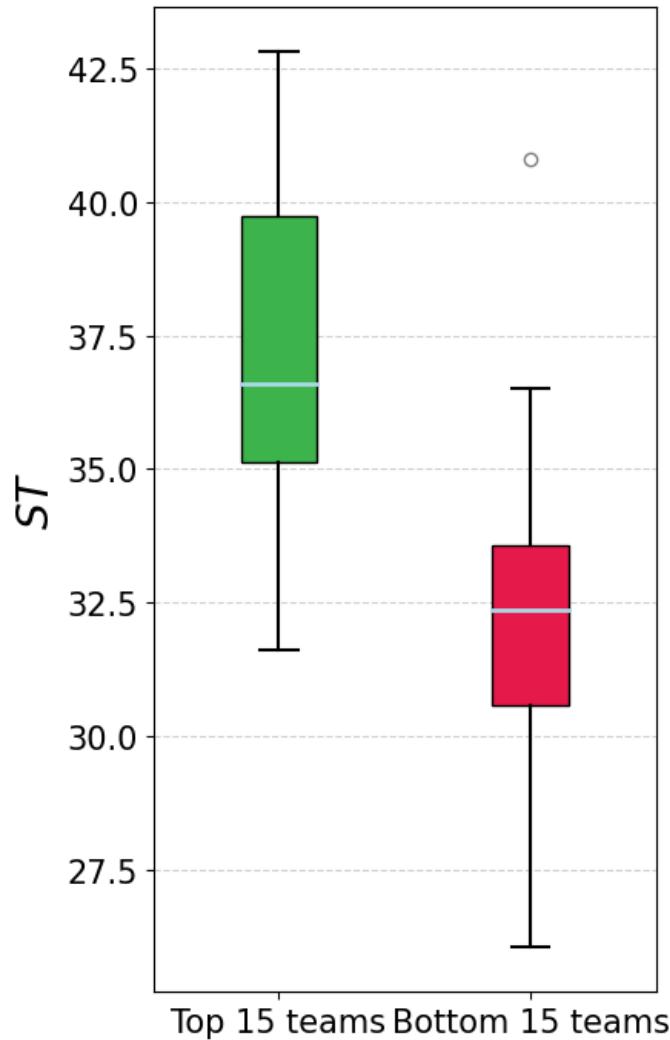


Figure 3.7: Box plots of the ST of the best 15 teams and of the bottom 15 teams.

According to this computation, it seems that our sparsity hypothesis is a reasonable assumption to make moving forward.

We will now use the ST to predict whether buying a player will improve the performance of a team. For our analysis, we will use a team called FC Barcelona as an example.

We now compute the ST of the Vietoris-Rips filtration of FC Barcelona, simulating the addition of a new player from the Spanish league. For each player, we compute the ST of FC Barcelona plus that player. After finishing this procedure, we create the following histogram which encapsulates this information:

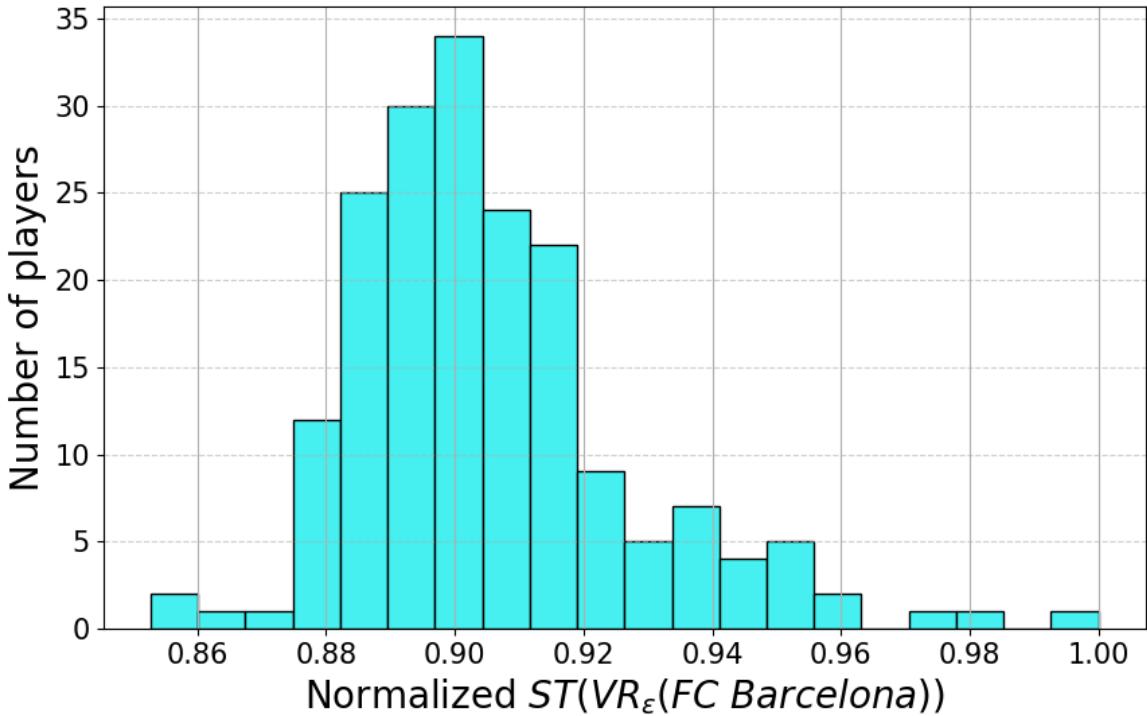


Figure 3.8: FC Barcelona histogram of normalized ST when adding a new player from LaLiga.

We now focus on the two rightmost light blue rectangles in the histogram. The furthest rectangle represents Karim Benzema, arguably the best player in the Spanish league. Next to it, we find Gerard Gumbau, a player of one of the worst-performing teams whose salary is close to 1% of Benzema's. Initially, we suspected that our sparsity assumption, represented by the ST , was flawed. Nonetheless, strikingly, we found the article Montenegro (2023) from FC Barcelona's newspaper, asserting that the team was considering buying Gerard Gumbau.

It is time now to move on to the final chapter of this thesis, where we will summarize our contributions and discuss future lines of research.

Conclusions

In this thesis, even though we have described and discussed multiple topics, we believe our most valuable contributions have been introducing a comprehensive explanation of the mathematical ideas behind UMAP and presenting three football applications: understanding how different features impact salaries, creating a graph to visualize and cluster players, and predicting the performance of new acquisitions. The astonishing insights derived from these applications exhibit the potential of category theory and TDA as foundational backbones of applied tools.

As for the future, we identify several organic lines of research to follow, particularly in applying various ML tools. As a first step, building a football dataset using computer vision seems like a natural direction. Then, the use of ML embeddings can help draw outstanding conclusions and strengthen our predictive power. Moreover, since players were locally represented, finding a global representation for them is a sensible next step. An analytic approach to solving this problem could involve defining a function from one league to another and finding some sort of “inverse”.

Finally, we think that virtually every discipline is entering a new era where most decisions will be made using data analysis, and visualization will play a fundamental role. We hope that the combination of artificial intelligence and robust mathematics can leverage human skills and help transform the world into a better place.

Bibliography

- Andrews, B. 2017, Lectures on Differential Geometry
- Anscombe, F. J. 1973, The american statistician, 27, 17
- Bombadil, T. 2013, How to draw a coffee cup, <https://tex.stackexchange.com/questions/145223/how-to-draw-a-coffee-cup>, accessed: 20-02-2024
- Csardi, G., & Nepusz, T. 2006, InterJournal, Complex Systems, 1695 (Link)
- Eilenberg, S., & MacLane, S. 1945, Transactions of the American Mathematical Society, 58, 231
- FBREF. 2023, FBREF, <https://www.fbref.com>, accessed: 23-03-2024
- Harris, C. R., et al. 2020, Nature, 585, 357–362
- Hofer, C., Kwitt, R., Niethammer, M., & Uhl, A. 2017, Advances in neural information processing systems, 30
- Hotelling, H. 1933, Journal of educational psychology, 24, 417
- Hunter, J. D. 2007, Computing in science & engineering, 9, 90
- Inc., P. T. 2015, Collaborative data science (Link)
- Jackson, A. 2019
- Johnstone, P. T. 2002, Sketches of an Elephant: A Topos Theory Compendium: Volume 2, Vol. 2 (Oxford University Press)
- Knight, P. A. 2008, SIAM Journal on Matrix Analysis and Applications, 30, 261
- Kuhn, H. W. 1955, Naval research logistics quarterly, 2, 83
- LeCun, Y., Cortes, C., & Burges, C. 2010, ATT Labs [Online]. Available: <http://yann.lecun.com/exdb/mnist>, 2
- Leinster, T. 2014, Basic category theory, Vol. 143 (Cambridge University Press)

- Lewis, M. 2004, Moneyball: The art of winning an unfair game (WW Norton & Company)
- Love, E. R., Filippenko, B., Maroulas, V., & Carlsson, G. 2021, arXiv preprint arXiv:2101.05778
- Mac Lane, S. 2013, Categories for the working mathematician, Vol. 5 (Springer Science & Business Media)
- McInnes, L. 2018, in Proceedings of the 17th Python in Science Conference, SciPy (Austin, TX, USA: SciPy) (Link)
- McInnes, L. 2018, How UMAP Works, https://umap-learn.readthedocs.io/en/latest/how_umap_works.html, accessed: 09-02-2024
- McInnes, L., Healy, J., & Melville, J. 2018, arXiv preprint arXiv:1802.03426
- McKinney, W., et al. 2010, in Proceedings of the 9th Python in Science Conference, Vol. 445, Austin, TX, 51–56
- Montenegro, C. 2023, Gumbau, sorprendente nuevo candidato del Barça para el medio-centro, accessed: 2023-04-23 (Link)
- Nanda, V. 2021, Computational Algebraic Topology - Lecture Notes
- Nichols, J. 2024, in preparation
- Pearson, K. 1901, The London, Edinburgh, and Dublin philosophical magazine and journal of science, 2, 559
- Pedregosa, F., et al. 2011, Journal of machine learning research, 12, 2825
- Riehl, E. 2017, Category theory in context (Courier Dover Publications)
- Schutz. 2010, Anscombe's quartet, <https://en.wikipedia.org/wiki/File:Anscombe>
- Sharland, P. 2022, Opinion: Sassuolo forward Domenico Berardi is most underrated player in Europe and it's not that close, accessed: 2023-04-18 (Link)
- Shlens, J. 2014, arXiv preprint arXiv:1404.1100
- Singh, G., Mémoli, F., Carlsson, G. E., et al. 2007, PBG@ Eurographics, 2, 091
- Smale, S. 1957, Proceedings of the American mathematical society, 8, 604
- Spivak, D. I. 2009, Preprint, 4

The GUDHI Project. 2015, GUDHI User and Reference Manual (GUDHI Editorial Board) (Link)

Van Rossum, G., & Drake, F. L. 2009, Python 3 Reference Manual (Scotts Valley, CA: CreateSpace)

van Veen, H. J., Saul, N., Eargle, D., & Mangham, S. W. 2019, Journal of Open Source Software, 4, 1315 (Link)

Waskom, M., et al. 2017, mwaskom/seaborn: v0.8.1 (September 2017) (Link)

Xiao, H., Rasul, K., & Vollgraf, R. 2017, arXiv preprint arXiv:1708.07747

Zhang, Y., et al. 2021, Horticulture research, 8